

Ramon Hugo de Souza

**MAPEAMENTO DE QUALIDADE DE EXPERIÊNCIA
(QOE) ATRAVÉS DE QUALIDADE DE SERVIÇO (QOS)
FOCADO EM BASES DE DADOS DISTRIBUÍDAS**

Tese submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina para a obtenção do Grau de doutor em Ciência da Computação.
Orientador: Prof. Dr. Mario Antonio Ribeiro Dantas

Florianópolis

2017

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

de Souza, Ramon Hugo
Mapeamento de Qualidade de Experiência (QoE)
Através de Qualidade de Serviço (QoS) Focado em
Bases de Dados Distribuídas / Ramon Hugo de Souza ;
orientador, Mario Antonio Ribeiro Dantas -
Florianópolis, SC, 2016.
138 p.

Tese (doutorado) - Universidade Federal de Santa
Catarina, Centro Tecnológico, Programa de Pós
Graduação em Ciência da Computação, Florianópolis,
2016.

Inclui referências.

1. Ciência da Computação. 2. Qualidade de
Experiência (QoE). 3. Qualidade de Serviço (QoS). 4.
Arquiteturas de Bases de Dados Distribuídas. 5.
Serviços em Nuvem. I. Dantas, Mario Antonio Ribeiro.
II. Universidade Federal de Santa Catarina.
Programa de Pós-Graduação em Ciência da Computação.
III. Título.

Ramon Hugo de Souza

**MAPEAMENTO DE QUALIDADE DE EXPERIÊNCIA
(QOE) ATRAVÉS DE QUALIDADE DE SERVIÇO (QOS)
FOCADO EM BASES DE DADOS DISTRIBUÍDAS**

Esta Tese foi julgada aprovada para a obtenção do Título de “doutor em Ciência da Computação”, e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina.

Florianópolis, 15 de fevereiro 2017.

Profa. Carina Friedrich Dorneles, Dra.
Coordenadora do Programa

Banca Examinadora:

Prof. Mario Antonio Ribeiro Dantas, Dr.
Universidade Federal de Santa Catarina
Orientador

Profa. Renata Galante, Dra.
Universidade Federal do Rio Grande do Sul

Prof. Nelson Francisco Favilla Ebecken, Dr.
Universidade Federal do Rio de Janeiro (videoconferência)

Prof. José Leomar Todesco, Dr.
Universidade Federal de Santa Catarina

Prof. Roberto Willrich, Dr.
Universidade Federal de Santa Catarina

Prof. Ronaldo dos Santos Mello, Dr.
Universidade Federal de Santa Catarina



In memorium of Anaís, desaparecida em
27 de outubro de 2015.

AGRADECIMENTOS

Agradeço, inicialmente, ao professor Ronaldo dos Santos Mello, membro da banca e professor a primeiro me acolher junto ao programa de pós-graduação em ciência da computação, com a primeira turma de doutorado, em 2012, e quem me encaminhou subsequentemente à orientação do professor Mario Dantas, não apenas por estes fatos, mas também pelo apoio na confecção desta tese estruturalmente. Agradeço em especial à meu orientador, e amigo a tantos anos, professor Mario Antonio Ribeiro Dantas, pela confiança e liberdade disponibilizada para realização da pesquisa que resultou nesta tese. Agradeço também ao professor Frank Siqueira, pelas revisões detalhadas que permitiram publicações de alto nível, e a seu orientando, e colega quem acompanho o trabalho desde os tempos de graduação, Paulo Arion, pela ajuda em resolução de problemas quando aplicados à bases de dados distribuídas, que resultou em publicação em conjunto de ambos. Ao professor Roberto Willrich, também membro da banca, agradeço pela introdução aos conceitos de qualidade de serviço, com quem trabalhei em meu primeiro projeto junto à UFSC em 2001, chamado projeto UCER, o qual resultou na tese de doutorado do professor Roberto Dias, hoje professor junto ao IFSC, e um de meus professores durante o curso técnico em eletrônica nos tempos em que o instituto ainda se chamava ETFSC.

Mathematical reasoning may be regarded rather schematically as the exercise of a combination of two facilities, which we may call intuition and ingenuity.

(Alan Turing, 1938)

RESUMO

A falta de conceitualização congruente sobre qualidade de serviço (QoS) para bases de dados (BDs) foi o fator que impulsionou o estudo resultante nesta tese. A definição de QoS como uma simples verificação de se um nó corre risco de falha devido ao número de acessos, como faziam, na época do levantamento bibliométrico desta tese, alguns sistemas comerciais, era uma simplificação exagerada para englobar um conceito tão complexo. Outros trabalhos que dizem lidar com estes conceitos também não são exatos, em termos matemáticos, e não possuem definições concretas ou com qualidade passível de utilização ou replicação, o que torna inviável sua aplicação ou mesmo verificação. O foco deste estudo é direcionado à bases de dados distribuídas (BDDs), de maneira que a conceitualização aqui desenvolvida é também compatível, ao menos parcialmente, com modelos não distribuídos de BDs. As novas definições de QoS desenvolvidas são utilizadas para se lidar com o conceito correlacionado de qualidade de experiência (QoE), em uma abordagem em nível de sistema focada em completude de QoS. Mesmo sendo QoE um conceito multidimensional, difícil de ser mensurado, o foco é mantido em uma abordagem passível de mensuramento, de maneira a permitir que sistemas de BDDs possam lidar com autoavaliação. A proposta de autoavaliação surge da necessidade de identificação de problemas passíveis de autocorreção. Tendo-se QoS bem definida, de maneira estatística, pode-se fazer análise de comportamento e tendência comportamental de maneira a se inferir previsão de estados futuros, o que permite o início de processo de correção antes que se alcance estados inesperados, por predição estatística. Sendo o objetivo geral desta tese a definição de métricas de QoS e QoE, com foco em BDDs, lidando com a hipótese de que é possível se definir QoE estatisticamente com base em QoS, para propósitos de nível de sistema. Ambos os conceitos sendo novos para BDDs quando lidando com métricas mensuráveis exatas. E com estes conceitos então definidos, um modelo de recuperação arquitetural é apresentado e testado para demonstração de resultados quando da utilização das métricas definidas para predição comportamental.

Palavras-chave: Qualidade da Experiência (QoE). Qualidade de Serviço (QoS). Arquiteturas de Bases de Dados Distribuídas. Serviços em Nuvem. Big Data. Confiabilidade. Disponibilidade. Facilidade de Manutenção. Perspectiva do Usuário.

ABSTRACT

The hitherto lack of quality of service (QoS) congruent conceptualization to databases (DBs) was the factor that drove the initial development of this thesis. To define QoS as a simple verification that if a node is at risk of failure due to memory over-commitment, as did some commercial systems at the time that was made the bibliometric survey of this thesis, it is an oversimplification to encompass such a complex concept. Other studies that quote to deal with this concept are not accurate and lack concrete definitions or quality allowing its use, making infeasible its application or even verification. Being the focus targeted to distributed databases (DDBs), the developed conceptualization is also compatible, at least partially, with models of non-distributed DBs. These newfound QoS settings are then used to handle the correlated concept of quality of experience (QoE) in a system-level approach, focused on QoS completeness. Being QoE a multidimensional concept, hard to be measured, the focus is kept in an approach liable of measurement, in a way to allow DDBs systems to deal with self-evaluation. The idea of self-evaluation arises from the need of identifying problems subject to self-correction. With QoS statistically well-defined, it is possible to analyse behavior and to identify tendencies in order to predict future states, allowing early correction before the system reaches unexpected states. Being the general objective of this thesis the definition of metrics of QoS and QoE, focused on DDBs, dealing with the hypothesis that it is possible to define QoE statistically based on QoS, for system level purposes. Both these concepts being new to DDBs when dealing with exact measurable metrics. Once defined these concepts, an architectural recovering model is presented and tested to demonstrate the results when using the metrics defined for behavioral prediction.

Keywords: Quality of Experience (QoE). Quality of Service (QoS). Distributed Database Architectures. Cloud Services. Big Data. Reliability. Availability. Serviceability. User's Perspective.

LISTA DE FIGURAS

Figura 1	Histograma de comportamento da pesquisa em <i>Web of Knowledge: Web of Science & Derwent Innovations Index Databases</i> . Tópico selecionado: (“Quality of Experience” OR QoE) de 2001 a 2013 em http://apps.webofknowledge.com obtido em 6 de Janeiro de 2014.....	51
Figura 2	Histograma de comportamento de pesquisa em <i>Scopus</i> . Título de artigo, resumo, palavras chave selecionado: (“Quality of Experience” OR QoE) de 2001 a 2013 em http://apps.webofknowledge.com obtido em 8 de Janeiro de 2014.....	55
Figura 3	Distribuição normal bem comportada da base <i>Scopus</i> na área de pesquisa de <i>ciência da computação</i> . Resultados para o tópico: “Quality of Experience” or QoE de 2001 a 2013 obtidos em 8 de Janeiro de 2014.....	60
Figura 4	Distribuição normal bem comportada da base <i>Scopus</i> nas áreas de pesquisa de <i>ciência da computação</i> e <i>engenharia</i> combinadas. Resultados para o tópico: “Quality of Experience” or QoE de 2001 a 2013 obtidos em 8 de Janeiro de 2014.....	61
Figura 5	Distribuição normal <i>não bem comportada</i> da base <i>Web of Knowledge</i> na área de pesquisa de <i>ciência da computação</i> . Resultados para o tópico: “Quality of Experience” or QoE de 2001 a 2013 obtidos em 6 de Janeiro de 2014.....	61
Figura 6	Distribuição normal <i>não bem comportada</i> da base <i>Web of Knowledge</i> nas áreas de pesquisa de <i>ciência da computação</i> e <i>engenharia</i> combinadas. Resultados para o tópico: “Quality of Experience” or QoE de 2001 a 2013 obtidos em 6 de Janeiro de 2014.	62
Figura 7	Desvio padrão acumulado para 2σ	75
Figura 8	Modelo arquitetural RAS, como publicado em (SOUZA et al., 2016), com implementação de troca de mensagens utilizando o modelo multianel com exemplo de tolerância a uma falta.....	91
Figura 9	O modelo arquitetural estendido com 10 nós iniciais no <i>pool</i> provedor de serviços (em azul e amarelo) e 4 nós iniciais no <i>pool</i> de réplicas dormentes (em vermelho), com um máximo de 16 nós em uma implementação de troca de mensagens em multianel com 2 anéis completos, com tolerância a uma falta, e 1 semianel auxiliar de réplicas dormentes.	96
Figura 10	Recursos requeridos e disponíveis do modelo otimista...	108

Figura 11 Recursos requeridos e disponíveis do modelo balanceado.	109
Figura 12 Recursos requeridos e disponíveis do modelo não-otimista.	110
Figura 13 Recursos requeridos e disponíveis do modelo não-otimista aprimorado.....	111
Figura 14 Variação de recursos individuais de nós no modelo otimista para alcançar os valores de recursos requisitados.....	112
Figura 15 Variação de recursos individuais de nós no modelo balanceado para alcançar os valores de recursos requisitados	112
Figura 16 Variação de recursos individuais de nós no modelo não-otimista para alcançar os valores de recursos requisitados.....	113
Figura 17 Variação de recursos individuais de nós no modelo não-otimista aprimorado para alcançar os valores de recursos requisitados.....	113
Figura 18 Número de nós no pool provedor de serviços no modelo otimista em resposta aos recursos requisitados	114
Figura 19 Número de nós no pool provedor de serviços no modelo balanceado em resposta aos recursos requisitados	114
Figura 20 Número de nós no pool provedor de serviços no modelo não-otimista em resposta aos recursos requisitados.....	115
Figura 21 Número de nós no pool provedor de serviços no modelo não-otimista aprimorado em resposta aos recursos requisitados ...	115

LISTA DE TABELAS

Tabela 1	Pesquisa sobre <i>Web of Knowledge: Web of Science & Derwent Innovations Index Databases</i> . Tópico selecionado: (“Quality of Experience” OR QoE) de http://apps.webofknowledge.com realizada em 6 de Janeiro de 2014.....	50
Tabela 2	Pesquisa sobre <i>Web of Knowledge: Web of Science & Derwent Innovations Index Databases</i> . Tópico selecionado: (“Quality of Experience” OR QoE) AND (Database OR “Distributed Architecture” OR “Big Data”) de http://apps.webofknowledge.com realizada em 12 de fevereiro de 2014.	52
Tabela 3	Pesquisa sobre <i>Web of Knowledge: Web of Science & Derwent Innovations Index Databases</i> . Tópico selecionado: (“Quality of Experience” OR QoE) AND (Database OR “Distributed Architecture” OR “Big Data”) resultados para 2012-2013 em <i>ciênc. da comp.</i> ou <i>eng.</i> de http://apps.webofknowledge.com obtidos em 10 de fev. de 2014.	52
Tabela 4	Pesquisa sobre <i>Web of Knowledge: Web of Science & Derwent Innovations Index Databases</i> . Tópico selecionado: (“Quality of Experience” OR QoE) AND (Database OR “Distributed Architecture” OR “Big Data”) resultados para 2001-2011 em <i>ciência da computação</i> ou <i>engenharia</i> de http://apps.webofknowledge.com obtidos em 10 de fevereiro de 2014.....	53
Tabela 5	Artigos relevantes correlacionando <i>QoE</i> e <i>arquitecturas de BDDs</i> encontrados na base de dados <i>Web of Knoledge</i> em 6 de Janeiro de 2014.....	53
Tabela 6	Pesquisa no <i>Scopus</i> . Título de artigo, resumo e palavras chave selecionadas: (“Quality of Experience” OR QoE) em http://www.scopus.com realizada em 8 de Janeiro de 2014.....	55
Tabela 7	Pesquisa no <i>Scopus</i> . Título de artigo, resumo, palavras chave selecionado: (“Quality of Experience” OR QoE) AND (Database OR “Distributed Architecture” OR “Big Data”) em http://www.scopus.com realizada em 12 de Fevereiro de 2014.....	56
Tabela 8	Pesquisa no <i>Scopus</i> . Título de artigo, resumo, palavras chave selecionadas: (“Quality of Experience” OR QoE) AND (Database OR “Distributed Architecture” OR “Big Data”) resultados de 2013 em <i>ciência da computação</i> e <i>engenharia</i> em http://www.scopus.com obtidos em 12 de Fevereiro de 2014.	57

Tabela 9 Pesquisa no <i>Scopus</i> . Título de artigo, resumo, palavra chave selecionado: (“Quality of Experience” OR QoE) AND (Database OR “Distributed Architecture” OR “Big Data”) resultados de 2012 em <i>ciência da computação e engenharia</i> em http://www.scopus.com obtidos em 12 de Fevereiro de 2014.	58
Tabela 10 Pesquisa no <i>Scopus</i> . Título de artigo, resumo, palavras chave selecionado: (“Quality of Experience” OR QoE) AND (Database OR “Distributed Architecture” OR “Big Data”) resultados de 2001-2011 em <i>ciência da computação e engenharia</i> em http://www.scopus.com obtidos em 12 de fevereiro de 2014.	59
Tabela 11 Garantias de QoS do tipo: Capacidades R&Q, como propostas na tese.	71
Tabela 12 Garantias de QoS do tipo: Capacidades RAS, como propostas na tese.	72
Tabela 13 Fatores que influenciam <i>QoE</i> e suas medidas mais importantes (GONG et al., 2009).	72
Tabela 14 Garantias de QoS do serviço de inserção.	74
Tabela 15 Garantias de QoS do serviço de atualização.	77
Tabela 16 Garantias de QoS do serviço de deleção.	78
Tabela 17 Garantias de QoS para serviço de seleção.	79
Tabela 18 Fila de distribuição de requisições de operações.	98
Tabela 19 Artigos publicados em ordem cronológica.	137

LISTA DE ABREVIATURAS E SIGLAS

ANS	Acordo de Nível de Serviço
BD	Base de Dados
BDD	Base de Dados Distribuída
CX	<i>Customer Experience</i>
DBMS	<i>Database Management System</i>
IPC	Indicador de Performance Chave
NoSQL	<i>Not Only SQL</i>
ONS	Objetivos em Nível de Serviço
QoE	<i>Quality of Experience</i>
QoS	<i>Quality of Service</i>
QX	<i>Quality of Experience</i>
RAD	Requisição de Acesso a Dados
RAS	<i>Reliability/Availability/Serviceability</i>
R&Q	<i>Resources and Quality</i>
SGBD	Sistema de Gerenciamento de Bases de Dados
SLA	<i>Service Layer Agreement</i>
TI	Tecnologia da Informação
UX	<i>User Experience</i>

LISTA DE SÍMBOLOS

σ	Desvio Padrão: Medida de dispersão existente em relação à média
O	<i>Big O</i> : Notação assintótica descrevendo limitação superior
Ω	<i>Big Omega</i> : Notação assintótica descrevendo limitação inferior
Θ	<i>Big Theta</i> : Notação assintótica descrevendo limitação superior e inferior
o	<i>Small O</i> : Notação assintótica descrevendo dominação superior
ω	<i>Small Omega</i> : Notação assintótica descrevendo dominação inferior
\sim	Da ordem de: Notação assintótica descrevendo igualdade

SUMÁRIO

1	INTRODUÇÃO	25
1.1	CONTEXTUALIZAÇÃO DO PROBLEMA: EXPERIÊNCIA	28
1.2	FOCO ANALÍTICO DA PESQUISA	30
1.3	OBJETIVOS	32
1.3.1	Objetivo Geral	32
1.3.2	Objetivos Específicos	32
1.4	ORGANIZAÇÃO DA TESE	33
2	CONTEXTUALIZAÇÃO	35
2.1	QUALIDADE DE SERVIÇO (QOS)	35
2.2	QUALIDADE DE EXPERIÊNCIA (QOE)	38
2.3	A ABORDAGEM DE MAPEAMENTO DE QUALIDADE DE EXPERIÊNCIA ATRAVÉS DE QUALIDADE DE SERVIÇO	42
2.4	CONCEITUAÇÃO DE NOSQL E ARQUITETURAS DE BASES DE DADOS DISTRIBUÍDAS	42
2.4.1	Complexidade de Operações em Arquiteturas de Bases de Dados Distribuídas	44
2.4.2	Arquiteturas de Bases de Dados Distribuídas e Qualidade de Experiência	45
3	ANÁLISE BIBLIOMÉTRICA	47
3.1	WEB OF KNOWLEDGE	47
3.2	SCOPUS	47
3.3	TÉCNICAS E PROCEDIMENTOS	48
3.4	RESULTADOS ESTATÍSTICOS E ANÁLISE	48
3.4.1	Resultados da base de dados Web of Knowledge	49
3.4.1.1	Análise dos resultados combinados do Web of Science & Derwent Innovations Index da base de dados Web of Knowledge	49
3.4.2	Resultados da base de dados Scopus	54
3.4.2.1	Análise dos resultados sobre a base de dados Scopus	54
3.4.3	Análise estatística de interesse de pesquisa	60
3.5	DETALHES DA REVISÃO BIBLIOGRÁFICA	62
3.5.1	Artigo: Quality of experience in distributed databases	62
3.5.2	Artigo: Providing quality of experience for users: the next DBMS challenge	65

4	MODELAGEM ARQUITETURAL.....	67
4.1	MODELO DE MAPEAMENTO TEÓRICO DE QUALIDADE DE EXPERIÊNCIA ATRAVÉS DE QUALIDADE DE SERVIÇO PARA BASES DE DADOS DISTRIBUÍDAS.....	67
4.1.1	Serviço de inserção: garantias de QoS e medição de QoE	73
4.1.2	Serviço de atualização: garantias de QoS e medição de QoE	76
4.1.3	Serviço de delegação: garantias de QoS e medição de QoE	78
4.1.4	Serviço de seleção: garantias de QoS e medição de QoE.....	79
4.1.5	Efeito das equações sobre QoE do sistema e usuário	80
4.1.6	Implicações do modelo de equações	82
4.2	MODELO ARQUITETURAL DE RECUPERAÇÃO PARA BASES DE DADOS DISTRIBUÍDAS FOCADO EM R.A.S.	84
4.2.1	Análise das métricas	86
4.2.2	Modelo Arquitetural RAS	90
4.2.2.1	Procedimentos Algorítmicos	92
4.3	MODELO ARQUITETURAL ESTENDIDO COM FOCO EM RECURSOS DE SISTEMA.....	94
4.3.1	Modelo de Completude de Recursos	95
4.3.1.1	Modos de Operação	100
4.3.1.2	Procedimentos Algorítmicos	101
4.4	CONSIDERAÇÕES SOBRE A MODELAGEM	102
4.5	AMBIENTE DE EXPERIMENTAÇÃO	103
4.5.1	Algoritmo de Simulação	103
4.5.2	Considerações sobre o Ambiente	105
5	ANÁLISE EXPERIMENTAL.....	107
5.1	RESULTADOS DE SIMULAÇÃO	107
5.2	LIMITAÇÕES E CASOS DE FALHA.....	116
6	CONCLUSÕES E TRABALHOS FUTUROS	117
	REFERÊNCIAS	121
	ANEXO A – Trabalhos Publicados	137

1 INTRODUÇÃO

A ausência de definições congruentes de *qualidade* para *bases de dados distribuídas* (*BDDs*), aliada aos conceitos já bem estabelecidos na área de *redes e comunicação*, e sua natureza holística, levaram este trabalho a iniciar com definições conceituais, com base na literatura existente, para demonstrar a importância da definição de *qualidade de experiência* (*QoE*) com foco voltado à área de *BDDs*.

Sem definições exatas de *qualidade*, sejam elas temporais ou de acuidade relativa à replicação, não existe maneira de se avaliar operações de *BDDs* de maneira objetiva. Sendo este um ponto fundamental para o desenvolvimento de *sistemas gerenciadores de bancos de dados* (*SGBDs*) de maior eficiência.

O conceito abstrato de *QoE*, focado em avaliar como o usuário *se sente* em relação ao sistema, é desenvolvido à um conceito mais maduro neste trabalho.

Sendo a definição de *QoE* um problema até então não bem fundamentado neste campo, a importância da caracterização do problema se demonstrou melhor abordada quando se lidando com o conceito em suas origens. Um ponto de grande importância deste trabalho sendo a demonstração de que uma melhor definição de *qualidade de serviço* (*QoS*) - um conceito originalmente focado em avaliar pontualmente um serviço contratado -, em termos de média e desvio padrão, dando intervalos de confiança específicos, pode ser utilizada para dar garantias de *QoE*, não apenas em *BDDs*, mas até mesmo em sistemas relacionados à *redes e comunicação* e *sistemas distribuídos* em geral.

Esta, até então, falta de definição de *QoE* para *BDDs*, parece ter tido sua origem em um vazio nas definições do conceito em si, o que permite interpretações fortemente holísticas, as quais não se adequam à abordagens centradas em comportamento de sistemas, como a necessária à resoluções relacionadas a *BDDs*. Em uma pesquisa exploratória inicial, a falta de definições matemáticas e, mais precisamente, estatísticas, relacionadas à *QoS* também foi encontrada, tornando difícil para um sistema lidar com avaliações focadas em autocorreção para prever, e mesmo se recuperar de, comportamentos inesperados.

Tornando, esta falta de conceituação bem definida relacionada à garantias de *QoS*, impossível a consideração de *QoE* em uma abordagem centrada em sistemas, de maneira a ser um grande desafio a ser suplantado a predição sistemática de comportamento futuro para evitar que situações inesperadas sejam alcançadas. Como resultado, sendo

necessária a definição matemática inicial de *QoS* para utilização com novas definições de garantias de *QoE*, com foco estatístico, aplicadas à área de *BDDs*.

Estes vazios conceituais tem sua origem também na associação de *BDDs* com a vasta gama de tecnologias conhecida como *NoSQL*. Como tais tecnologias tem como base apenas o fato de não trabalharem somente com restrições *SQL*, fica difícil a fundamentação de *QoS* de maneira generalizada para *BDDs*.

As definições propostas neste trabalho são necessárias para se lidar com avaliação de serviços aprimorada, e para se ser capaz de prover serviços de *BDDs* melhorados. Já que nenhum modelo matemático confiável havia até então sido proposto, as primeiras definições deste trabalho surgem como uma nova abordagem para melhorar a utilização de serviços relacionados à *BDDs*. De maneira que ao se aplicar estas avaliações de *QoE* a cenários envolvendo serviços oferecidos por *BDDs*, é possível o desenvolvimento de mecanismos de avaliação e autocorreção de performance de *sistemas distribuídos* em geral.

A base deste trabalho surge do questionamento:

É possível se avaliar qualidade de experiência de maneira sistemática quando se lidando com sistemas de bases de dados distribuídos?

Tentando responder a esta pergunta, um modelo arquitetural de completude temporal é proposto, lidando com a utilização de equações de avaliação, onde é apresentado o modelo teórico de mapeamento de *QoE* através de *QoS*, com foco em *BDDs*. Este modelo arquitetural inovador lida com soluções distribuídas, tanto horizontais quanto verticais, com foco na escolha de procedimentos mais adequados a serem seguidos, com base nos conceitos de confiabilidade e disponibilidade.

De maneira a demonstrar a fiabilidade de soluções com base neste modelo arquitetural, é proposta uma generalização passível de aplicação não apenas a *BDDs*, mas a qualquer modelo de sistemas distribuídos, para facilitar as demonstrações via experimentação, com quatro alternativas focadas em disponibilidade de recursos apresentadas para o modelo de recuperação arquitetural. Já que os conceitos como apresentados são focados em sistemas distribuídos, o modelo arquitetural apresentado pode ser naturalmente estendido para outros serviços distribuídos além das quatro operações de *BDDs* apresentadas inicialmente.

O modelo inicial apresentado lida basicamente com garantias de completude de tempo, que é o caminho natural a ser seguido quando se

considerando avaliação de operações de *bases de dados* (*BDs*). Com o foco em uma avaliação mais realista, em termos de conceitos de arquiteturas distribuídas, o foco em completude temporal é posteriormente modificado para completude de recursos, de maneira a também incluir o conceito primeiramente apresentado de completude temporal. O motivo é claramente evitar mapeamentos que poderiam gerar dificuldades em garantir a fiabilidade do modelo, afinal completude temporal precisaria ser mapeada para completude de recursos de qualquer maneira.

Os exemplos são apresentados na arquitetura *multianel*, com tolerância a uma falta, inspirados no exemplo apresentado em (MARANDI; PRIMI; PEDONE, 2012), de maneira a ilustrar a possibilidade de se lidar com um provedor de serviços de dois níveis, sem exceder o número de mensagens de atualização. São apresentados também resultados empíricos relativos aos quatro modelos definidos com base em completude de recursos.

A metodologia utilizada neste estudo tem como início fundamental o levantamento bibliométrico, demonstrando a falta de abordagem congruente sobre o assunto até o momento de tal análise junto ao meio científico, seguido de definições teóricas de *QoS* para posterior fundamentação de definições de *QoE*. Uma vez definidas métricas de *QoE* com base em *QoS*, é apresentado um *modelo arquitetural de recuperação*, fundamentado em análise comportamental utilizando as métricas definidas. Por fim, é apresentado um estudo empírico sobre o modelo generalizado, juntamente com as conclusões subsequentes do mesmo.

Como inovação são apresentadas as definições de *QoS* para *BDDs*, definidas em duas famílias, e o mapeamento de definições de *QoE*, em nível de sistema, como completude de tais definições de *QoS*. Os dois modelos de avaliação para recuperação, tanto temporal quanto de disponibilidade de recursos, sendo também novas abordagens introduzidas à área de *BDDs*. E com o modelo de disponibilidade de recursos são também introduzidas quatro abordagens para se lidar com o gerenciamento de tais recursos.

Sendo o estudo apresentado de fundamentação teórica, as demonstrações realizadas visam somente apresentar a aplicação dos fundamentos aqui definidos, não se baseando tal estudo em verificação comportamental por experimentação. Neste modelo, experimentos servem para verificação de comportamento de sistemas, de maneira a enquadrar o comportamento no modelo e permitir a tomada de decisões com base em tal comportamento. Em suma, este é um trabalho científico de fundamentação, lidando com estudo de casos para ilustrar aplicações possíveis e caminhos a serem trilhados futuramente.

1.1 CONTEXTUALIZAÇÃO DO PROBLEMA: EXPERIÊNCIA

A definição de *qualidade de experiência do cliente*, de acordo com FY/c em (CLOUD, 2013) é, em resumo, “a diferença entre expectativas e percepções para um serviço”, sendo, desta maneira, uma medida subjetiva que não é apenas difícil de se quantificar mas também difícil de se definir em termos de parâmetros, medindo a experiência do cliente com um serviço especificado.

Esta é uma definição que pode ser abstraída em um conceito mais maduro removendo-se a variável cliente, já que as definições originais de *qualidade* vêm normalmente associadas a conceitos relacionados a redes, onde o *cliente*, a quem o serviço é provisto, é normalmente um *usuário da internet* esperando a entrega usual de serviço por *melhor esforço*.

Mesmo em conceitualizações *relacionadas a redes*, a ideia de se *satisfazer um cliente* é considerada um objetivo abstrato a ser alcançado. Como um sistema não tem como saber o que o cliente quer *per se*, e isto pode envolver questões psicológicas abstratas, lidar com novas conceitualizações a serem alcançadas não é um bom ponto de partida.

Tendo-se preestabelecido tais restrições, o conceito de *experiência* a ser avaliada pode ser então encapsulado em *nível de sistema*. Não apenas o sistema deve ser capaz de avaliar seu nível de completude de serviços segundo um nível de *qualidade* esperado, definido explicitamente em um *acordo de nível de serviço*¹ (ANS), mais comumente referenciado em inglês como *service layer agreement* (SLA), essencial para recuperação de estados inesperados, mas também ser capaz de avaliar a *experiência de usuário sob demanda* e evitar excesso de consumo de recursos que possa levar a estados de indisponibilidade.

Mesmo sabendo-se que “*QoE* é um conceito que não é limitado apenas ao uso de sistemas ou serviços” (CALLET; MÖLLER; PERKIS, 2013), a avaliação pode ser parcialmente definida como tal para prover a um sistema, ou serviço, a habilidade de autoavaliar sua completude, como apresentado na definição 1, publicada originalmente em (SOUZA; DANTAS, 2015).

Definição 1 *O conceito multidimensional de QoE pode ser parcialmente avaliado como completude de QoS, se a QoS for estatisticamente bem definida, para propósitos de avaliação em nível de sistema.*

¹ANS: é o acordo firmado com o cliente, o qual descreve o serviço contratado, suas metas de nível de serviço e os papéis e responsabilidades das partes envolvidas no acordo.

Uma questão importante sendo o estabelecimento de conceitos estatísticos para a *qualidade* a ser alcançada. Quando se lida com médias e desvios padrão, e especialmente com a combinação dos dois, garantias estatísticas definidas em *ANS* ajudam a evitar conceitualizações abstratas que podem levar a problemas de medição, e então garantias, do serviço a ser provisto segundo um acordo.

O conceito de *qualidade de experiência*, com suas origens em conceitos de redes, normalmente referenciado somente como *QoE* ou *QX*, tem seu foco principal na experiência de serviço como um todo. Este conceito é normalmente considerado uma avaliação menos conceitual que o já bem definido conceito de *experiência de usuário* (*user experience*), referenciado como *UX*, focado em interfaces de software, e o conceito de *experiência de cliente* (*customer experience*), normalmente referenciado como *CX*, com foco em suporte (CLOUD, 2014).

Um conceito que parece considerar avaliações abstratas relacionadas à *experiência* é baseado em medidas intangíveis. Sistemas devem ser capazes de avaliar considerações matemáticas ao invés de lidar com considerações psicológicas abstratas. Se a análise for mantida na área de *ciência da computação*, e portanto em considerações e conceitos de matemática discreta, considerações psicológicas não devem ser consideradas, afinal não fazem sequer parte da mesma área de conhecimento. O objetivo deve ser garantir um *ANS*, não conceitos abstratos de *felicidade*.

Aspectos técnicos focados em *satisfação do cliente e entrega de serviço* a que se referem a parâmetros de *QoS* parecem ser possivelmente lidados, de acordo com (LAGHARI et al., 2011), por plataformas comerciais. Entretanto, existem fatores como “problemas psicológicos subjetivos e aspectos cognitivos humanos que são tipicamente desconSIDERADOS e determinam diretamente a *QoE*” (LAGHARI et al., 2011). Tais considerações podem ser evitadas com a conceitualização de *QoE* como um mapeamento direto para definições de *QoS* em um *ANS*.

Para serem medidas, as definições originais de *QoE* parecem necessitar de sistemas especializados específicos, que possam de alguma maneira avaliar estes fatores desconSIDERADOS. Estes *sistemas de QoE*, de acordo com (LAGHARI et al., 2011), são sistemas que tentam lidar com o mensuramento de métricas que afetam diretamente a percepção do usuário como um parâmetro de qualidade. Em resumo, (LAGHARI et al., 2011) define que *QoE* provê uma avaliação das expectativas de usuários baseada em como estes usuários percebem um serviço específico, medindo sua satisfação.

Mantendo-se o foco em *BDDs*, estas considerações podem ser

ligadas estritamente a sistemas baseados em *ANS* e os problemas envolvendo comportamento inesperado, pela visão do usuário, podem ser caracterizados apenas como uma má escolha de *ANS*, um problema que pode ser facilmente corrigido adquirindo-se um *ANS* diferente.

Tal como definido na explicação teórica da conexão entre os conceitos de *QoE* e *QoS*, também em (LAGHARI et al., 2011), *QoE* pode ser considerada como um mapeamento direto, porém *difuso*, de como o usuário percebe um serviço em termos de *parâmetros de QoS*. Vale ressaltar que, uma vez que *parâmetros de QoE* são baseados em comportamento e expectativas humanas, é difícil se garantir um certo nível de precisão neste processo de mapeamento. Este é um problema que desaparece quando se considera *sistemas* ao invés de *usuários*.

Nesta tese é proposta uma maneira de se lidar com o mapeamento destas métricas de comportamento esperado em nível de sistema, não considerando o usuário diretamente, mas o acordo definido segundo os *conceitos de QoS* apresentados neste estudo.

Já que muitos dos estudos sobre *QoE-QoS* são encontrados na área de pesquisa de redes, esta falta de fundamentação em *BDs*, e especialmente em *arquiteturas distribuídas*, aponta para uma área em aberto com muitos tópicos interessantes que ainda precisam ser desenvolvidos a um nível mais maduro. Neste estudo, definições de *métricas de QoS* para *BDDs* são realizadas em uma tentativa de se mensurar *QoE* em *nível de sistema* para então se poder oferecer garantias comportamentais.

Para se lidar com tais conceitos, que tem suas origens na área de redes, e recentemente também discutidos em uma abordagem *centrada em redes* com foco em multimídia (CALLET; MÖLLER; PERKIS, 2013; MÖLLER; RAAKE, 2014), é naturalmente inferido que a variação de atraso, normalmente referida como *jitter* na terminologia de redes, em um serviço provido, é um fator que influencia diretamente o nível de *QoE* como percebido pelo usuário consumindo este serviço. Neste contexto, o desvio padrão é considerado como um fator influenciacional quando avaliando a experiência do usuário, sendo o mesmo explicitamente definido como parte dos *parâmetros do ANS* para nossos propósitos.

1.2 FOCO ANALÍTICO DA PESQUISA

O foco inicial deste estudo está na análise estatística da utilização do termo *QoE*, especialmente em publicações nas áreas de pesquisa

de *ciência da computação e engenharia*, para embasar os conceitos definidos subsequentemente. Um dos pontos interessantes da análise comportamental é o levantamento do uso relevante do termo no campo acadêmico, encontrando-se o ano em que se iniciou a utilização do termo em publicações científicas, bem como comportamento normal indicando o ano de maior utilização do termo nas referidas áreas.

O foco do estudo na área de *ciência da computação* é autoexplicativo, mas foi também percebido que o termo e os conceitos eram comumente utilizados em publicações na área de pesquisa referenciada como engenharia. Esta é a razão do foco de pesquisa se dar nessas duas grandes áreas.

Depois de se determinar o momento em que o termo começou a ser utilizado com a relevância esperada, o estudo se aprofunda no assunto de *QoE* de acordo com a área de *arquiteturas de BDDs*. Com estas considerações, o volume de dados encontrado foi restrito, e a análise individual de artigos mostrou a quase inexistência de trabalhos relacionados. Os dois principais artigos encontrados, (COSTA; FURTADO, 2011, 2013), escritos pelos mesmos autores, e mesmo com o primeiro deles mostrando resultados preliminares, o segundo fala do assunto como um problema ainda em aberto.

Em conclusão, estes dois artigos parecem mostrar direções possíveis, mas que ainda precisam ser melhor definidas para que possam ser implementadas na forma de *sistemas de QoE* lidando com decisões na *área de BDDs*.

Cabe notar que, quando considerando serviços nas nuvens, especialmente quando se considera migração de serviços para nuvens, *QoE* é comumente mencionado como um conceito crucial que carece de estudos envolvendo o ponto de vista do usuário (VANDENBROUCKE; MOOR; MAREZ, 2013). Além disso, é mencionado na pesquisa exploratória (VANDENBROUCKE; MOOR; MAREZ, 2013) que disponibilidade, acessibilidade e compatibilidade são considerados fatores de crucial importância, assim como aspectos relacionados a custo, privacidade e segurança.

Tais conceitos parecem bem fundamentados, entretanto não existem estudos exploratórios na *área de BDDs*. Esta falta de estudos parece deixar uma porta em aberto sem convergência de conceitos ou implementações ainda a serem fundamentadas.

1.3 OBJETIVOS

O objetivo inicial deste estudo é o de fundamentar conceitos e métricas de *QoS* para *BDs*, com foco específico em *BDDs*, de maneira que estes conceitos e métricas possam ser então utilizados como parâmetros para avaliação de *QoE* em nível de sistema.

Uma vez definidas métricas de *QoS*, estas servem de base para a análise de *QoE* como completude das mesmas. De maneira que, tendo-se as métricas, pode-se prever comportamento futuro estatisticamente e evitar que estados e comportamentos inesperados sejam alcançados.

Sendo a definição de *QoE* para *BDDs* a base fundamental das proposições apresentadas nesta tese.

Assim, parte-se da simples hipótese de que uma melhor definição estatística de serviços em um *ANS*, utilizando médias e desvios padrão sobre definições de *QoS*, pode ser utilizada para se medir a completude dos serviços como esperados/contratados, e portanto a *QoE* em um ponto de vista focado em *sistemas*.

Uma vez tendo-se tais definições bem fundamentadas, são então apresentados sistemas de avaliação com proposições de correção para tais sistemas.

1.3.1 Objetivo Geral

Como objetivo geral tem-se a definição de *QoE* para *BDDs*, com base em *QoS*, para se mensurar a completude do *ANS*, sendo estes conceitos novos para tal área. E com este mensuramento torna-se então possível se lidar com predição comportamental, o que também permite modificação do sistema em tempo de execução, de maneira a tornar possível se evitar que estados inesperados sejam alcançados com a possibilidade de desenvolvimento de novos *SGBDs* aprimorados.

1.3.2 Objetivos Específicos

A definição congruente de *QoS* para *BDDs* é o ponto de partida para que se possa definir *QoE* com base na mesma. Mas, uma vez tendo a possibilidade de se mensurar o comportamento do sistema, tem-se a possibilidade de se verificar se o mesmo está se comportando como o esperado, e mais, uma vez não estando se comportando como o esperado pode-se tentar corrigir o comportamento do mesmo.

A predição com base em análise comportamental se tornou foco para modificação de sistemas reais em tempo de execução, com um modelo arquitetural, baseado em uma das famílias de métricas de *QoS* apresentada, se tornando objetivo subsequente, e então sendo desenvolvido tal modelo para permitir análise e recuperação antes que estados inesperados sejam alcançados.

Um modelo generalizado de avaliação comportamental é apresentado ao final do estudo, com foco em completude de recursos, sendo aplicável à sistemas distribuídos em geral. Junto a este é também apresentado um estudo empírico sobre quatro abordagens consideradas.

1.4 ORGANIZAÇÃO DA TESE

Neste capítulo inicial abordou-se os porquês deste estudo, com uma breve contextualização do problema de maneira a se explicitar o objeto de estudo, apresentando também o foco analítico da pesquisa.

No capítulo 2 apresenta-se a conceitualização necessária para embasar o que se espera alcançar, apresentando a origem dos conceitos utilizados de *qualidade de serviço* e *qualidade de experiência*, bem como a interconexão envolvendo ambos os conceitos.

A análise bibliométrica, demonstrando a relevância do trabalho, é apresentada no capítulo 3, onde também apresenta-se os trabalhos que mais se aproximam do tema até então.

No capítulo 4 define-se os conceitos de *qualidade de serviço*, necessários para posterior definição de métricas de *qualidade de experiência* lidando com completude dos conceitos previamente apresentados. O modelo teórico apresentado na seção 4.1 serve de base para fundamentação da sequência do trabalho.

Um primeiro modelo arquitetural de recuperação é apresentado na seção 4.2, com foco em métricas de confiabilidade e disponibilidade. Sendo o mesmo estendido na seção 4.3, com foco em recursos de sistema, de maneira a facilitar a experimentação.

Para obtenção de resultados demonstrando o comportamento de sistemas, na seção 4.5 é apresentado um ambiente de experimentação generalizado para lidar com o modelo apresentado na seção 4.3.

No capítulo 5 apresenta-se um estudo empírico explicativo sobre as quatro abordagens selecionadas para o modelo apresentado na seção 4.3 com a utilização do ambiente apresentado na seção 4.5.

As conclusões são então apresentadas no capítulo 6.

2 CONTEXTUALIZAÇÃO

Sendo o assunto foco deste estudo muito específico, *QoE* aplicada a *arquiteturas de BDDs*, e centrado em conceitos abstratos que não são bem definidos no contexto de *BDs*, este estudo teve que, primeiro, preencher uma lacuna de conceituação para dar base ao caminho a ser percorrido.

A necessidade de *conceitualização de QoS* insta a partir da abordagem escolhida para a *definição de QoE* como mapeamento parcial em parâmetros de *QoS*, como exposto no artigo “*QoE aware service delivery in distributed environment*” (LAGHARI et al., 2011).

A ausência de artigos em *conceituação de QoE* na área de *arquitetura de BDDs* e a existência de várias abordagens para o conceito em si revelam a necessidade de conceitualização e demonstram a significância da abordagem selecionada.

Os conceitos de *arquiteturas de BDDs* e *NoSQL* são apresentados para contextualizar este problema de associação de conceitos abstratos com definições vagas de tecnologia, como a aplicação do conceito de *QoE* à modelos computacionais.

2.1 QUALIDADE DE SERVIÇO (QOS)

No relatório técnico “*The relationship between quality of service and business metrics: Monitoring, notification and optimization*” (WOLTER; MOORSEL, 2001), “o efeito da degradação de qualidade de serviço na rentabilidade de serviços eletrônicos” é discutida, sendo do interesse deste estudo a caracterização de *métricas de QoS* como taxa de transferência, atraso e disponibilidade, mas com média e desvio padrão de tempo de resposta combinados como garantias de disponibilidade.

De maneira similar a muitas publicações relacionadas a *QoS*, esta lida com estes conceitos em sua concepção original, como métricas relacionadas a redes. Neste relatório técnico em especial o foco é mantido em serviços eletrônicos (*e-services*).

Em uma consideração mais voltada à *QoE*, é sabido que, no que diz respeito a um *e-service*, em uma consideração do parâmetro de atraso, o usuário não irá aceitar mais que de oito a dez segundos quando lidando com um sítio eletrônico de *e-commerce* (BHATTI; BOUCH; KUCHINSKY, 2000; BOUCH; BHATTI; KUCHINSKY, 2000). Isto é até mesmo

um pouco pior que a variação de atraso se o atraso for mantido dentro desta quantidade de tempo. É sabido que a variação do atraso deixa o usuário desconfortável com a utilização de um serviço, mas um atraso grande demais torna a usabilidade do serviço inaceitável, especialmente em se considerando *e-commerce*.

O impacto dos *efeitos de QoS* em negócios lidando com relacionamentos dinâmicos, apresentado em (WOLTER; MOORSEL, 2001), já é um elo fraco com as *métricas de QoE* que serão discutidas nas seções subsequentes.

Baseando-se em *considerações de QoS*, taxa de transferência, atraso e disponibilidade são listadas como as principais métricas em (WOLTER; MOORSEL, 2001). Mas, em uma definição mais rígida apresentada em (SHENKER; PARTRIDGE; GUERIN, 1997), um *serviço com garantias* é definido como um serviço que “provê limites firmes e matematicamente demonstráveis em datagramas de enfileiramento de atraso fim a fim”.

Em (SHENKER; PARTRIDGE; GUERIN, 1997), garantias de serviço, focadas em definições para *QoS*, são apenas sobre atraso e largura de banda, não se considerando disponibilidade, já que as definições utilizadas não são sobre aplicações de *e-commerce*, como em (WOLTER; MOORSEL, 2001). Em termos de rede puros, a abstração de serviços consumidos em nível de software não está envolvida nas garantias dadas. As definições em (SHENKER; PARTRIDGE; GUERIN, 1997) são sobre mecanismos que buscam conseguir uma reserva garantida de recursos, com o modelo de especificação de serviço descrito em (SHENKER; WROCLAWSKI, 1997).

A definição conceitual de *métricas de QoS* como métricas com garantias matematicamente prováveis, como definido em (SHENKER; PARTRIDGE; GUERIN, 1997), dá uma conceitualização mais palpável para definição destas *métricas de QoS*.

Com base em tais definições, é facilmente percebido que os principais *conceitos de QoS* compreendem: (i) taxa de transferência, um conceito relacionado a largura de banda e taxa de bits; (ii) atraso, um conceito bem definido por si só; (iii) variação de atraso, uma variável correlacionada a atraso com origem nos conceitos de redes; e (iv) disponibilidade, que é uma consideração sobre probabilidade de falha, diretamente relacionada à probabilidade de perda de pacotes e/ou erros de *bits* em termos estritos de redes.

Além disso, em uma conceitualização mais *direcionada a nuvens*, dada em (ERL; PUTTINI; MAHMOOD, 2013), os serviços chave de métricas de qualidade são listados como:

- **Taxa de Disponibilidade:** “Percentagem de tempo de serviço em operação; medida como o tempo total de serviço em operação contra o tempo total” e expressa em percentagens;
- **Duração de Interrupção:** “Duração de uma única interrupção”; medida por data e hora em que a interrupção iniciou e terminou, expressa em horas e minutos;
- **Tempo Médio entre Falhas:** “Tempo esperado entre falhas de serviços consecutivas”; medido pelo período normal de operação e o número de falhas e expresso como o número médio de dias;
- **Taxa de Confiabilidade:** “Percentagem de sucesso de resultado de serviços sob situações preestabelecidas”; medida pelo número total de respostas bem sucedidas, expressa como percentagens;
- **Capacidade de Rede:** “Características mensuráveis de capacidade de rede”; medidas por largura de banda, taxa de transferência em *bits* por segundo, expressa como número de *megabits* por segundo;
- **Capacidade de Armazenamento de Dispositivo:** “Características mensuráveis de capacidade de armazenamento de dispositivo”, medida e expressa em tamanho de armazenamento em *gigabytes*;
- **Capacidade do Servidor:** “Características mensuráveis de capacidade de servidor”, medida e expressa como número de CPUs, frequência de CPU em GHz, RAM e espaço de armazenamento em GBs;
- **Capacidade de Aplicação Web:** “Características mensuráveis de capacidade de aplicação Web”, medida e expressa como requisições por minuto;
- **Tempo de Inicialização de Instância:** “Tempo necessário para inicializar uma nova instância”, medida pela data e hora que a instância está inicializada e a data e hora do início da requisição; expressa em minutos;
- **Tempo de Resposta:** “Tempo requerido para realizar uma operação síncrona”, medida pela data e hora de resposta e o número total de requisições, expressa como médias em milissegundos;
- **Tempo de Conclusão:** “Tempo requerido para completar uma tarefa assíncrona”, medido pela data de requisição até a data de resposta e o número total de requisições; expresso como médias em segundos;

- **Escalabilidade de Armazenamento (Horizontal):** “Alterações permitidas de capacidade de dispositivos de armazenamento em resposta a aumento de cargas de trabalho”, medida e expressa em tamanho de armazenamento em *gigabytes*;
- **Escalabilidade de Servidor (Horizontal):** “Mudanças de capacidade de servidor permitidas em resposta a aumento de cargas de trabalho”, medidas e expressas como número de servidores virtuais em *pool* de recursos;
- **Escalabilidade de Servidor (Vertical):** “Flutuações de capacidade de servidor permitidas em resposta a flutuações de carga de trabalho”, medida e expressa como número de CPUs e quantidade de RAM em *gigabytes*;
- **Tempo Médio para Comutação:** “Tempo esperado para se concluir uma troca completa de um serviço em falha para uma instância replicada em uma área em diferente localização geográfica”, medido pela data e hora de falha e o número total de falhas; expresso em minutos;
- **Tempo Médio de Recuperação do Sistema:** “Tempo esperado para um sistema flexível executar uma recuperação completa de uma falha de serviço”, medido pela data e hora da recuperação até a data e hora da falha e o número total de falhas; expresso em minutos.

Todas as métricas acima são métricas de qualidade que proveem *garantia de serviços* com métricas mensuráveis - matematicamente demonstráveis - que fornecem limites firmemente em serviços fim a fim como definido em (SHENKER; PARTRIDGE; GUERIN, 1997), e também proveem um modelo mais bem definido para a *abordagem de QoS* aplicada a *arquiteturas de BDDs*, mesmo não sendo todas as métricas aplicáveis à *QoS* como necessário para a *abordagem em BDDs*.

Estas *métricas chave de qualidade de serviço* são usadas como base para definir-se as métricas de qualidade do modelo desenvolvido, apresentado no capítulo 4.

2.2 QUALIDADE DE EXPERIÊNCIA (QOE)

O conceito de *qualidade de serviço* é bem definido em vários artigos *relacionados à marketing* (LEWIS; BOOMS, 1983; GRONROOS, 1984; PARASURAMAN; ZEITHAML; BERY, 1985, 1988, 1994), e pode ser concisamente definido como “o resultado da comparação que clientes fazem

entre suas expectativas sobre um serviço e sua percepção da maneira como o serviço foi realizado” (CARUANA, 2002). Esta conceitualização se liga ao conceito usado para *QoE*, e não *QoS*, em *ciência da computação*, e provavelmente tem suas origens nestes conceitos de marketing qualificando a experiência do usuário sobre o serviço.

Este *link* com conceitos de marketing pode ser visualizado na definição de (ISO9241-210, 2010), indicando que a *experiência do usuário* seria definida como “as percepções e respostas de uma pessoa que resultam do uso ou antecipação do uso de um produto, sistema ou serviço”.

Como dito no título do artigo (MORRIS; TURNER, 2001) - “*Assessing users’ subjective quality of experience with the world wide web: an exploratory examination of temporal changes in technology acceptance*” -, *QoE* é basicamente definida como uma métrica subjetiva. O termo *qualidade de experiência subjetiva dos usuários*, como diz o título, foi usado nas primeiras publicações por volta de 2001, quando o termo começou a se tornar popular no meio acadêmico, mostrando este sentimento sobre a subjetividade do conceito. O conceito de *aceitação de usuário*, citado em (MORRIS; TURNER, 2001) como uma “uma medida chave para avaliação de tecnologia da informação em pesquisas relacionadas a TI” com uma parte na definição de determinantes para este tipo de tecnologia, enfatiza a importância de se definir “ganho descritivo de informação e garantias de aumento de utilidade” para um serviço. Mesmo o foco deste estudo não sendo na *world wide web*, como em (MORRIS; TURNER, 2001), a conceitualização de *aceitação de usuário* como uma medida chave para avaliar serviços genericamente pode também ser definida como o coração dos *conceitos de QoE* como considerados no presente estudo - onde a conceitualização irá na verdade se desenvolver para uma mais genérica *aceitação de sistema* com o progresso da pesquisa.

Em uma abordagem mais focada em *ciência da computação*, os usuários, em uma consideração genérica de aspectos de serviço, são considerados como os que tem alguma necessidade que pode ser provista, de alguma maneira, pelos serviços oferecidos. A provisão de serviços implica em uma interação entre o usuário e a infraestrutura que provê este serviço. O usuário espera que esta relação se dê de forma transparente. De modo a alcançar isto, alguns esforços podem ser tomados, com *QoE* usualmente lidando com os conceitos de *perfis de serviço* e/ou *habilitadores de contexto*.

Perfis de serviço (ABRAMOWICZ et al., 2006) é um conceito que ajuda a prover uma melhor experiência do usuário pré-selecionando

valores de configurações para uma definição customizada de serviço. Uma vez que o perfil de serviço é selecionado, os serviços devem ser providos de acordo com as definições para tal perfil.

Habilitadores de contexto (RICHTER; BOHM, 2006) é uma solução proposta para se obter automaticamente informações de usuários, para então decidir, quando lidando com sistemas de tempo real, em como escolher a melhor maneira de entrega que se encaixe nas necessidades do usuário.

É interessante apontar que, em termos de conceitos de redes, *habilitadores de contexto* parece ser uma solução que assegura uma baixa variância de atraso na entrega de um serviço, independentemente de como o serviço será entregue.

A solução selecionada, seja *sob demanda* ou baseada na pré-seleção de *perfis*, é essencial para a aplicação dos *conceitos de QoE*, que precisam, de alguma maneira, ser mapeados para *definições de QoS* (LAGHARI et al., 2011).

A parte mais difícil de se *definir QoE* é a de inferir medidas de *qualidade de serviço* comparando-as com as também inferidas *medidas de experiência de usuário*. Como mencionado previamente, é muito difícil se lidar com métricas que afetam diretamente a percepção do usuário como *parâmetros de qualidade*.

Em (LAGHARI et al., 2012), *QoE* é considerada um conceito abstrato que, com base na análise de utilização de dados do usuário, pode ser utilizada para criar garantias diretamente em *parâmetros QoS*. Neste trabalho, *QoE* é definida como “um conjunto de fatores centrados em pessoas que proveem a avaliação de qualidade dos serviços, redes e dispositivos de usuário final”. É também afirmado que “*QoS* é uma abordagem centrada em tecnologia” e “falta-lhe satisfazer os requisitos hedônicos e estéticos humanos”.

Como definido em (LAGHARI et al., 2011), *QoE* é um recurso que fornece a avaliação de “expectativas, sentimentos, percepções, cognição e satisfação humanas com respeito a um particular produto, serviço ou aplicação”. Além disso, como mencionado em (LAGHARI et al., 2012), para se capturar *QoE*, alguns métodos subjetivos e objetivos são considerados. Os métodos subjetivos se baseiam em comentários sobre qualidade de indivíduos avaliados; e os métodos objetivos, como classificado em (LAGHARI; CRESPI; CONNELLY, 2012), são “(i) fatores técnicos objetivos que inferem *QoE* dos dados disponíveis de *QoS*, e (ii) fatores objetivos que estão relacionados aos sistemas psicológicos e cognitivos humanos”.

No estudo de (BROOKS; HESTNES, 2010), a importância de defini-

ções objetivas e quantitativas para *QoE* é discutida. A conceitualização de *QoE* com foco em requerimentos para sua medição é desenvolvida, e é argumentado que “a medição de *QoE* deve incluir medidas objetivas do processo e resultados de utilização”. É também mencionado que “medidas subjetivas devem ser coletadas como dados quantitativos ao invés de qualitativos como é atualmente feito, porque dados quantitativos habilitam análises combinadas de medidas objetivas e subjetivas para derivação de classificação de uma *QoE global*”. Esta *classificação global* é considerada importante para a otimização da comunicabilidade dos resultados focando em *audiências* específicas.

Três abordagens para “medição de *serviço de qualidade em redes* da perspectiva do usuário” são apresentadas com suas vantagens e desvantagens em (BROOKS; HESTNES, 2010): (i) Teste de percepção de *QoS* pelo usuário: onde a principal vantagem é o uso de medidas diretas do comportamento do usuário, e duas desvantagens críticas são apresentadas como “o foco do usuário final estando inteiramente no *domínio perceptivo*” (BROOKS; HESTNES, 2010), e o relacionamento entre o comportamento do usuário e fatores psicológicos inconscientes não sendo visualizado em opinião autorrelatada; (ii) Levantamento subjetivo de *QoE*: um estudo de comportamento de usuário não contando com a opinião do usuário, onde “*QoE* é inferida de medições técnicas indiretas” (BROOKS; HESTNES, 2010) com testes utilizados para identificar, e posteriormente validar, a relação inferida do comportamento do usuário com parâmetros técnicos; e (iii) Modelagem de qualidade de mídia: com modelos computacionais ditos objetivos “porque modelam parâmetros técnicos mensuráveis” (BROOKS; HESTNES, 2010; ITU-T.P.862, 2001) com a vantagem de ser centrado no usuário, e com as desvantagens de *velocidade e economia* por lidar com validação contínua contra dados de teste do usuário e as limitações de validação do modelo contra os dados de percepção do usuário - a já mencionada relação entre comportamento do usuário e fatores psicológicos inconscientes.

O estudo de (BROOKS; HESTNES, 2010) também apresenta medidas objetivas de desempenho de usuário como: precisão de conclusão de tarefa de usuário; ganho ou perda ao usuário; tempo necessário para atingir o objetivo; número de entradas do usuário; tempo entre entradas do usuário; número de turnos necessários para se comunicar; número de interrupções; número de ações simultâneas; e medidas subjetivas de desempenho de usuário como: efetividade; eficiência; satisfação; divertimento; presença social; e impressão de parceria de comunicação.

Todas estas medidas sendo muito construtivas para a *definição de QoE* de maneira geral, mas ainda não precisas o suficiente para a

abordagem *focada em BDDs* requerida por este estudo.

2.3 A ABORDAGEM DE MAPEAMENTO DE QUALIDADE DE EXPERIÊNCIA ATRAVÉS DE QUALIDADE DE SERVIÇO

Na literatura, especialmente na literatura relacionada a redes, sobre definições de *qualidade*, alguns autores fazem referência a *QoE* como um mapeamento direto de como o usuário percebe *QoS* (LAGHARI et al., 2011). O principal problema, quando considerando um *sistema de autoavaliação* que pode utilizar métricas para se recuperar de estados inesperados, considerando-se os trabalhos relacionados sobre *QoE* de (CLOUD, 2013; LAGHARI et al., 2011; CLOUD, 2014; VANDENBROUCKE; MOOR; MAREZ, 2013; LAGHARI et al., 2012), é que métricas impalpáveis não são úteis.

Estes trabalhos relacionados consideram *QoE* como uma avaliação *holística*, o que não tem uso do ponto de vista de sistemas. *Como o usuário se sente* não significa nada para o sistema. Saber se o usuário está satisfeito não significa que o sistema está trabalhando de acordo com o *ANS* contratado. Afinal, as únicas variáveis que importam, considerando-se *como o usuário se sente*, são aquelas sobre estabilidade de serviço, e então as previstas com garantias de desvio padrão associado.

Por causa disso, a fim de ser capaz de se utilizar métricas para um *sistema de autocorreção*, é primeiro necessário se definir métricas de uma maneira não holística. Mais explicitamente, definições de *métricas de QoE* mapeadas para *métricas palpáveis de QoS* são de extrema importância para a abordagem escolhida.

2.4 CONCEITUAÇÃO DE NOSQL E ARQUITETURAS DE BASES DE DADOS DISTRIBUÍDAS

A emergência de tecnologias de *arquiteturas distribuídas* alavancou-se da necessidade de se lidar com a integração de um grande volume de dados corporativos heterogêneos armazenados em diferentes localizações espaciais, sendo também a promoção de reusabilidade associada com a possível redução dos custos de manutenção um impulsionador apontando para esta tecnologia como uma solução atrativa para se lidar com *arquiteturas de BDs*.

Por exemplo, no caso de desempenho, quando se considera um

sistema com a premissa de muito mais acesso à dados do que gravações, pode-se facilmente inferir que a mesma seria otimista em uma solução distribuída apenas se imaginando que a replicação não iria causar muito processamento ao se verificar dados em servidores separados, já que consideramos um sistema que lida com acesso massivo e gravação eventual. Da mesma maneira, é fácil se inferir que múltiplos servidores poderiam lidar com o acesso de vários usuários com um melhor desempenho de que apenas um.

É claro que existem contraexemplos, mas, quando se considera este tipo de solução, o problema original deve apontar para este tipo de arquitetura. De outra maneira, esta não seria a solução adequada para o problema proposto. Este tipo de tecnologia se foca em soluções trabalhando com *sistemas híbridos* que requerem distribuição horizontal, com mais servidores, e não vertical, com mais memória e capacidade de processamento, sendo esta uma tecnologia fácil de escalar, normalmente mais eficiente para leitura do que para escrita.

Uma consideração recorrente a este tipo de *BDs* é sempre a necessidade de existência de um *sistema de gerenciamento de dados* para manter o controle e prover acesso coordenado a diversas bases de dados interconectadas. O desenvolvimento destes *sistemas de gerenciamento de dados* é o passo estrutural que resultou na introdução deste tipo de tecnologia como é conhecida hoje.

Os tipos de controle de *sistemas de gerenciamento de dados* podem ser genericamente caracterizados como *logicamente centralizados* (SMITH et al., 1981) e *descentralizados* (HEIMBIGNER; MCLEOD, 1985), também chamados de *federados*, sendo o tipo de gerenciamento um determinante do grau de autonomia, que afeta a natureza do acoplamento entre as interfaces de esquema.

Destes dois conceitos, estabelecidos nos anos 80, vários modelos de *BDs* evoluíram, cada um lidando com as limitações do modelo de uma maneira particular. Os *logicamente centralizados* normalmente sacrificando a autonomia local, e os *federados* normalmente retendo a autonomia local de maneira a tornar difícil o controle do sistema de maneira global.

O surgimento de modelos mais complexos, como os *baseados em conhecimento*, e não convencionais, como os *orientados a objetos*, levou a reconsideração das perspectivas de *arquiteturas de BDDs*. Com estes *novos conceitos* introduzidos ao campo de *BDs*, é agora possível se lidar com modelagem de dados em alto nível, estritamente conectada com o software sendo modelado. As arquiteturas *clusterizadas*, como o ambiente apresentado em (PAPAZOGLU; BUBENKO; NORRIE, 1988),

combinam os dois paradigmas - *logicamente centralizado e federado*. Elas são híbridos entre estes dois tipos de modelos com suas vantagens particulares.

Estes modelos de *BDs* híbridos são hoje, em grande parte dos casos, classificados como modelos *não apenas SQL (NoSQL)*. O conceito de *NoSQL* engloba modelos de *BDs* que não necessitam de *SQL* para acessar dados, sendo estes modelos então não necessariamente relacionais.

A família de modelos *NoSQL* inclui uma série de submodelos¹, como armazenamento de documentos, objetos, chave-valor (cache, armazenamento, armazenamento eventualmente consistente e armazenamento ordenado), colunas largas e família de colunas, estrutura de dados, armazenamento de tuplas, grafos, multimodelo, multidimensional, multivalor, baseado em eventos e algumas soluções específicas de *grid* e soluções para nuvens.

2.4.1 Complexidade de Operações em Arquiteturas de Bases de Dados Distribuídas

Para se lidar com complexidade computacional, em termos de notações o , O , ω , Ω e Θ , considerando-se *arquiteturas de BDDs*, seria necessário um grande esforço para se computar as complexas funções que descreveriam de maneira correta a complexidade de operações.

As variáveis de interesse necessárias para se obter uma função genérica poderiam incluir número de nós processando informação, o número de núcleos desses nós, como eles se comunicam, os protocolos utilizados, as probabilidades de falhas em termos de armazenamento e memória, entre outras variáveis muito mais difíceis de se conceber, e medir, se o foco envolver performance.

A complexidade pode então ser mensurada considerando-se apenas o número de operações, para se evitar variáveis relacionadas a *hardware*, mas o número de replicações e como elas se comunicam e garantem consistência ainda são muito difíceis de se mensurar em termos de uma função genérica que descreva a complexidade de operações. E mesmo que isso pudesse ser feito, em termos de operações, quando feito para uma *BD* específica ainda não seria uma conceituação válida completamente sobre o comportamento de uma arquitetura genérica.

Publicações sobre complexidade de operações, considerando-se *arquiteturas de BDDs*, em geral consideram apenas a complexidade de

¹Parcialmente baseado em (DATABASES, 2014).

troca de mensagens entre os nós replicantes e não se aprofundam em complexidade computacional propriamente dita.

De maneira que neste estudo lidaremos apenas com considerações relativas à replicação de mensagens, com um modelo tolerante a faltas apresentado na seção 4.2 e sua evolução subsequente na seção 4.3.

2.4.2 Arquiteturas de Bases de Dados Distribuídas e Qualidade de Experiência

As garantias de *QoE* podem ser utilizadas para dar uma maior certeza de provimento segundo necessidades específicas de usuário, sendo previstas em termos de alocação de recursos, prioridade de execução, ou qualquer outro tipo de definição de garantias que sejam acordadas quanto ao provimento de serviços como contratados.

As definições usuais de *perfis de serviço* (ABRAMOWICZ et al., 2006) e *habilitadores de contexto* (RICHTER; BOHM, 2006), apresentadas na seção 2.2, podem ser utilizadas para *predefinir* ou definir *segundo necessidade*, respectivamente, os requisitos do usuário quanto ao serviço consumido, e então pode-se utilizar a implementação destas definições para prover o serviço segundo as necessidades contratadas.

A falta de estudos interconectando as áreas de pesquisa de *arquiteturas de BDDs* e *QoE* é muito preocupante, já que o tópico *QoE* parece ter atingido seu ponto máximo de interesse em 2012, como demonstrado na seção 3.4, e as únicas referências a *QoE* em *arquiteturas de BDDs* parecendo ser implementações de *QoS*, ainda não implementadas de maneira a prover serviços para usuários não experientes.

Entretanto, as definições apresentadas em (COSTA; FURTADO, 2011), categorizando suboperações em termos de prioridade, parecem ser um primeiro passo em direção à implementação de *QoE* para *arquiteturas de BDDs*.

Os resultados estatísticos de (COSTA; FURTADO, 2013) mostram um primeiro estudo sobre o ganho que pode ser atingido se aplicando *QoE* à *arquiteturas de BDDs*.

Estas primeiras definições, bem como os resultados estatísticos apresentados, serviram de base para as definições realizadas nesta tese. Porém a impossibilidade de se reproduzir os dados dos mesmos, bem como vários questionamentos sobre a abordagem selecionada pelos autores, levaram este trabalho a seguir em direções um pouco diferenciadas. Estes dois trabalhos são detalhados na seção 3.5.

3 ANÁLISE BIBLIOMÉTRICA

Um estudo bibliométrico foi realizado para obtenção de dados que pudessem ser utilizados na presente pesquisa sobre *QoE* suportada por *QoS*. Para tal, foram escolhidas as bases *Web of Knowledge* (REUTERS, 2013) e *Scopus* (ELSEVIER, 2013), ambas com reconhecimento científico e acessibilidade a pesquisadores, que serviram como base para a análise do *estado da arte* do tema.

Os dados foram obtidos destas bases no início de 2014, sendo que os artigos publicados no ano de 2014 não haviam sido ainda indexados. Sendo digno de nota que até o final de 2016 não haviam surgido novos resultados relevantes quanto ao tema abordado.

3.1 WEB OF KNOWLEDGE

A base *Web of Knowledge* lida com sua principal coleção de artigos, chamada *Web of Science*, com dados que datam desde 1945 até o presente, com 65 áreas explícitas; com o *Derwent Innovations Index*, com “valor adicionado por informação de patentes do *Derwent World Patent Index*, também com informações de citação de patentes” (REUTERS, 2013) do *Patents Citation Index* de 1963 até o presente; e com o *Journal Citation Reports*.

Esta pesquisa se foca nas áreas de interesse de *ciência da computação* e *engenharia*, mas também apresenta dados obtidos quando feita pesquisa sobre o que a ferramenta chama de *ciências tecnológicas* e todas as subáreas da base.

3.2 SCOPUS

A base *Scopus* lida com sua própria seleção de publicações, onde os tipos de fonte são especificados como publicações comerciais, revistas, anais de conferências e coleções de livros, lidando com 19 áreas de interesse.

Na pesquisa realizada nesta base também manteve-se o foco em *ciência da computação* e *engenharia*, e além dos dados obtidos de todas as bases também apresenta-se os dados do que a ferramenta chama de *ciências físicas*, que incluem: engenharia química, química, ciência da computação, ciências planetárias e da terra, energia, engenharia, ciên-

cias ambientais, ciência dos materiais, matemática e física, astronomia e fontes multidisciplinares (ELSEVIER, 2013).

3.3 TÉCNICAS E PROCEDIMENTOS

Após a análise dos dados crús, o estudo então iniciou uma análise de conteúdo dos dados obtidos com restrição às áreas de pesquisa de *ciência da computação e engenharia*, já que o volume de dados encontrado, de maneira genérica, fora muito extenso.

Independente do comportamento estatístico esperado ter sido encontrado, a análise dos tópicos foi restringida para uma melhor análise de conteúdo de acordo com sua relevância.

Infelizmente, o volume de dados associados com a discretização anual foi muito grande, não apenas no caso em que houve a indicação de comportamento inicial - não exatamente como o esperado em forma normal -, mas também no caso que parece ter demonstrado o comportamento real. E, mesmo espelhando os dados para inferência de comportamento futuro esperado, os testes de normalidade Shapiro-Wilk (SHAPIRO; WILK, 1965), Jarque-Bera (JARQUE; BERA, 1980) e D'Agostino (D'AGOSTINO, 1970) pareceram se perder, mesmo com os dados claramente tendendo a um comportamento estatístico normal.

Com as restrições finais aos dados a análise de citação não se tornou essencial, devido ao volume de dados, muito pequeno e diretamente relacionado, mas mesmo assim apresenta-se o número de citações na análise de publicações com o tópico de interesse específico.

Os procedimentos metodológicos selecionados se basearam na busca de um comportamento normal estatístico, com um ano claro sendo indicado como o ponto de maior interesse de pesquisa sobre o assunto. O comportamento encontrado indicou que a normal esperada já passou por seu máximo, apontando para uma possível falta de interesse neste assunto em particular pela comunidade de *banco de dados*, já que os resultados são quase inexistentes até o presente momento.

3.4 RESULTADOS ESTATÍSTICOS E ANÁLISE

Esta pesquisa estatística inicial começou com a definição de alguns parâmetros de pesquisa, para se restringir o espaço de busca sobre as duas bases selecionadas e então obter-se dados mais relevantes.

Com a expectativa inicial de se poder lidar com a análise de todos

os dados obtidos sobre todas as áreas de pesquisa nas bases selecionadas, os resultados foram inicialmente filtrados para se evitar resultados com conotação errônea e que estaria fora do foco real da pesquisa. Por esta razão, o termo *QX* em especial foi retirado das pesquisas, para evitar um grande volume de resultados não relacionados. O termo *QoE* introduziu alguns resultados não relacionados, mas não com um volume tão grande quanto o termo *QX*. Como era mais fácil se lidar com os resultados não relacionados introduzidos pelo termo *QoE*, foi decidido se manter o mesmo nas pesquisas, já que muitos artigos utilizam o termo para especificamente citar *qualidade de experiência*.

3.4.1 Resultados da base de dados Web of Knowledge

Por ser uma ferramenta muito detalhada e de fácil utilização, inicialmente pensou-se em lidar somente com o *Web of Knowledge* para a obtenção dos dados necessários para preencher os requisitos de pesquisa - os resultados nesta ferramenta pareciam ser mais detalhados que os da segunda ferramenta selecionada. Infelizmente, um comportamento inesperado, não seguindo distribuição normal, criou a necessidade de se lidar com mais de uma ferramenta para se inferir sobre o comportamento real da relevância dos termos pesquisados.

3.4.1.1 Análise dos resultados combinados do Web of Science & Derwent Innovations Index da base de dados Web of Knowledge

A combinação das duas principais bases de dados do *Web of Knowledge* foi considerada para esta pesquisa. Em um momento inicial foi considerado se lidar com as bases de maneira separada, devido a avisos de número duplicado de artigos nos resultados; mas isto foi corrigido evitando-se o uso do termo *QX*.

O comportamento estatístico normal que era esperado não foi encontrado aqui. Ao que parece devido a uma *perturbação* entre 2009 e 2011, razão pela qual esta pesquisa lidou com duas ferramentas de indexação de artigos.

A análise iniciou-se pela obtenção da definição de um ano inicial com utilização relevante do conceito *qualidade de experiência*. Para tal escolheu-se a realização das pesquisas sobre as bases *Web of Knowledge* com o parâmetro *topic* definido como ("*Quality of Experience*" OR *QoE*) para verificar-se as primeiras referências à *QoE*. Em seguida,

o parâmetro *topic* foi definido como (“*Quality of Experience*” OR *QoE*) AND (Database OR “Distributed Architecture” OR “Big Data”) para verificação de referências à *QoE* associada com o foco em *arquiteturas de BBDs*.

Com os resultados obtidos nesta primeira pesquisa sobre o *Web of Knowledge*, refinando pelas áreas de pesquisa de *ciência da computação* ou *engenharia*, apresentados na tabela 1, foi encontrada uma referência inicial em 1998 (MIELKE et al., 1998). Esta sendo apenas uma coincidência na utilização do termo, e uma referência em 2001 no artigo “Assessing users’ subjective quality of experience with the world wide web: an exploratory examination of temporal changes in technology acceptance” (MORRIS; TURNER, 2001), com um uso já aceitável do termo para os propósitos desejados.

Web of Knowledge: Bases de Dados Web of Science & Derwent Innovations Index					
Topic: (“Quality of Experience” OR QoE)					
Ano	Geral	Ciências Tecnológicas	Ciência da Computação	Engenharia	Ciênc. da Comp. ou Engenharia
1945-2014	1619	1152	567	705	948
2014	2	2	2	0	2
2013	301	183	84	120	151
2012	394	300	111	191	236
2011	251	180	89	117	158
2010	187	133	63	90	117
2009	189	155	102	95	131
2008	125	90	65	66	87
2007	54	46	26	18	36
2006	25	20	11	5	16
2005	15	9	5	0	5
2004	15	10	2	0	2
2003	8	6	5	1	5
2002	10	1	0	0	0
2001	4	2	1	1	1
1999-2000	8	5	0	0	0
1998	5	3	1	1	1
1971-1997	27	7	0	0	0

Tabela 1: Pesquisa sobre *Web of Knowledge: Web of Science & Derwent Innovations Index Databases*. Tópico selecionado: (“Quality of Experience” OR QoE) de <http://apps.webofknowledge.com> realizada em 6 de Janeiro de 2014.

Infelizmente, como pode ser visto na figura 1, depois de 2009 *engenharia* não só passou *ciência da computação*, mas o termo *QoE* não pareceu ser tão relevante para *ciência da computação* como para os outros campos de pesquisa - um comportamento completamente oposto ao esperado -, ou de alguma maneira o *Web of Knowledge* parece ter negligenciado as publicações indexadas em *ciência da computação* desde 2009.

Mas, deixando este comportamento não esperado de lado, é reconhecido que se pode encolher o volume de publicações encontradas

quando refinando a pesquisa para os propósitos específicos deste estudo, como demonstrado na tabela 2.

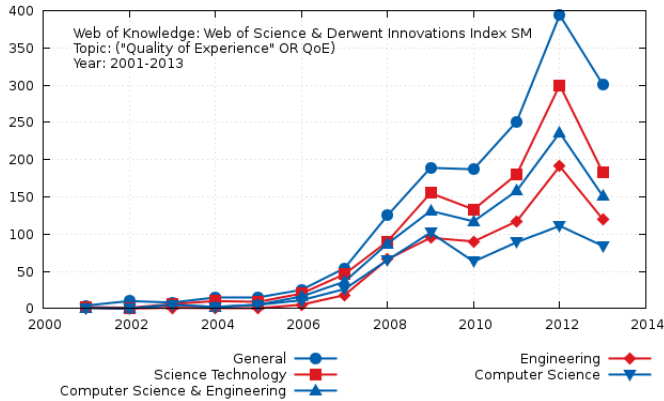


Figura 1: Histograma de comportamento da pesquisa em *Web of Knowledge: Web of Science & Derwent Innovations Index Databases*. Tópico selecionado: (“Quality of Experience” OR QoE) de 2001 a 2013 em <http://apps.webofknowledge.com> obtido em 6 de Janeiro de 2014.

Como pode ser visto, as tabelas 1 e 2 têm um comportamento muito similar mas, é claro, com um volume de dados muito menor na tabela 2. Dos 28 artigos encontrados apenas 2, (COSTA; FURTADO, 2011) e (COSTA; FURTADO, 2013), ambos de Rogério Luís de Carvalho Costa e Pedro Furtado, nas áreas de *ciência da computação* e *engenharia* são relevantes para este estudo.

Estes 28 artigos, apresentados nas tabelas 3 e 4, são sobre tópicos em geral relacionados a redes e *peer to peer* (como os artigos 06, 10, 18, 27 e 28), video, VoIP e sistemas de fala (como os artigos 04, 05, 08, 09, 12, 14, 15, 16, 20, 22, 24, 25 e 29), e tecnologias estereoscópicas de video e imagem (como os artigos 03, 07, 11, 19 e 23). Os dois artigos correlacionados lidando com conceitos de *BDDs* e *QoE*, listados na tabela 5, são detalhados na seção 3.5: *Detalhes da revisão bibliográfica*.

Os resultados sublinhados nas tabelas 3 e 4 foram encontrados simultaneamente nas bases *Scopus* e *Web of Knowledge*.

Web of Knowledge: Bases de Dados Web of Science & Derwent Innovations Index					
Topic: ("Quality of Experience" OR QoE) AND (Database OR "Distributed Architecture" OR "Big Data")					
Ano	Geral	Ciências Tecnológicas	Ciência da Computação	Engenharia	Ciênc. da Comp. ou Engenharia
1945-2014	56	35	16	19	29
2014	0	0	0	0	0
2013	16	10	4	4	6
2012	19	13	6	8	12
2011	7	4	2	3	4
2010	7	5	2	3	4
2009	2	0	0	0	0
2008	4	2	1	1	2
2007	1	1	1	0	1

Tabela 2: Pesquisa sobre *Web of Knowledge: Web of Science & Derwent Innovations Index Databases*. Tópico selecionado: ("Quality of Experience" OR QoE) AND (Database OR "Distributed Architecture" OR "Big Data") de <http://apps.webofknowledge.com> realizada em 12 de fevereiro de 2014.

Web of Knowledge: Bases de Dados Web of Science & Derwent Innovations Index		
n	Artigo	Citações
01	"Paired comparison-based subjective quality assessment of stereoscopic images" (LEE; GOLDMANN; EBRAHIMI, 2013)	4
02	"Subjective evaluation of stereoscopic image quality" (MOORTHY et al., 2013)	2
03	"Providing quality of experience for users: the next DBMS challenge" (COSTA; FURTADO, 2013)	0
04	"On quality of experience of scalable video adaptation (LI; CHEN; TAN, 2013)	1
05	Spatiotemporal no-reference video quality assessment model on distortions based on encoding" (ZERMAN et al., 2013a)	0
06	"Approach for service search and peer selection in P2P service overlays" (FIORESE; SIMOES; BOAVIDA, 2013)	0
07	"Rendering 3-D High dynamic range images: subjective evaluation of tone-mapping methods and preferred 3-D image attributes" (MAI et al., 2012)	0
08	"Guidelines for an improved quality of experience in 3-D TV and 3-D mobile displays" (XU; CORIA; NASIOPOULOS, 2012)	1
09	"QoE prediction model and its application in video quality adaptation over UMTS networks" (KHAN; SUN; IFEACHOR, 2012)	9
10	"QON: Quality of experience (QoE) framework for network services" (LAGHARI et al., 2012)	0
11	"A reputation based vertical handover decision making framework (R-VHDF)" (LOUKIL et al., 2012)	0
12	"QoE assessment of multimedia video consumption on tablet devices" (FLORIS et al., 2012)	0
13	"Quality of experience assessment for stereoscopic images" (QI et al., 2012)	0
14	"PNN-based QoE measuring model for video applications over LTE system" (HE et al., 2012)	0
15	"QoE analysis for scalable video adaptation" (LI; CHEN; TAN, 2012a)	0
16	"QoE-aware resource allocation for scalable video transmission over multiuser MIMO-OFDM systems" (LI; CHEN; TAN, 2012b)	0
17	"Comparison of stereoscopic technologies in various configurations" (FLIEGEL et al., 2012)	0
18	"Peer selection in P2P service overlays using geographical location criteria" (FIORESE; SIMOES; BOAVIDA, 2012)	0

Tabela 3: Pesquisa sobre *Web of Knowledge: Web of Science & Derwent Innovations Index Databases*. Tópico selecionado: ("Quality of Experience" OR QoE) AND (Database OR "Distributed Architecture" OR "Big Data") resultados para 2012-2013 em *ciênc. da comp.* ou *eng.* de <http://apps.webofknowledge.com> obtidos em 10 de fev. de 2014.

Web of Knowledge: Bases de Dados Web of Science & Derwent Innovations Index		
n	Artigo	Citações
19	"Quality of experience in distributed databases" (COSTA; FURTADO, 2011)	1
20	"A flexible QoE framework for video streaming services" (ALVAREZ et al., 2011)	0
21	"Objective metrics for quality of experience in stereoscopic images" (XING et al., 2011)	0
22	"A study of artificial speech quality assessors of VoIP calls subject to limited bursty packet losses" (JELASSI; RUBINO, 2011b)	0
23	"Proposed framework for evaluating quality of experience in a mobile, testbed-oriented living lab setting" (MOOR et al., 2010)	6
24	"New single-ended objective measure for non-intrusive speech quality evaluation" (MAHDI; PICOVICI, 2010)	2
25	"Comparison of approaches for instrumentally predicting the quality of text-to-speech systems" (MÖLLER et al., 2010)	1
26	"Temporal synchronization in stereoscopic video: influence on quality of experience and automatic asynchrony detection" (GOLDMANN; LEE; EBRAHIMI, 2010)	1
27	"Optimizing user QoE through overlay routing, bandwidth management and dynamic transcoding" (WIJNANTS et al., 2008)	0
28	"Dynamic QoS provisioning for ethernet-based networks" (ANGELOPOULOS et al., 2008)	0
29	"QoE monitoring platform for video delivery networks" (VERA; RODRIGUEZ-BOCCA; RUBINO, 2007)	1

Tabela 4: Pesquisa sobre *Web of Knowledge: Web of Science & Derwent Innovations Index Databases*. Tópico selecionado: ("Quality of Experience" OR QoE) AND (Database OR "Distributed Architecture" OR "Big Data") resultados para 2001-2011 em *ciência da computação* ou *engenharia* de <http://apps.webofknowledge.com> obtidos em 10 de fevereiro de 2014.

Web of Knowledge: Bases de Dados Web of Science & Derwent Innovations Index			
Artigo	Autores	Periódico	Ano
"Quality of experience in distributed databases" (COSTA; FURTADO, 2011)	Rogério Luís de Carvalho Costa e Pedro Furtado	Distributed and parallel databases	2011
"Providing quality of experience for users: the next DBMS challenge" (COSTA; FURTADO, 2013)	Rogério Luís de Carvalho Costa e Pedro Furtado	Computer - - -	2013

Tabela 5: Artigos relevantes correlacionando *QoE* e *arquiteturas de BDDs* encontrados na base de dados *Web of Knowledge* em 6 de Janeiro de 2014.

3.4.2 Resultados da base de dados Scopus

Da mesma maneira como foi feito para a base *Web of Knowledge*, o parâmetro “Article Title, Abstract and Keywords” foi definido como (“*Quality of Experience*” OR *QoE*) para verificação de primeiras referências a *QoE*. Após, o parâmetro “Article Title, Abstract and Keywords” foi definido como (“*Quality of Experience*” OR *QoE*) AND (Database OR “Distributed Architecture” OR “Big Data”) para verificação de referências a *QoE* com foco em *arquiteturas de BDDs*.

3.4.2.1 Análise dos resultados sobre a base de dados Scopus

Resultados da base *Scopus* apresentados na tabela 6, refinados por ciências físicas, mostram uma primeira referência em 1987 (REDD et al., 1987) que é, na verdade, uma referência à *Quasi-One-Electron* como *QoE*, e, refinando-se por *ciência da computação e engenharia*, são encontradas uma referência em 1989 (HOOGEBOOM, 1989), mas sobre desenvolvimento/redesenvolvimento de políticas para áreas de praias oceânicas, uma referência em 1998 (MIELKE et al., 1998) que é um artigo não relacionado já encontrado na pesquisa do *Web of Knowledge*, referências em três artigos também não relacionados em 2000: (SATO; VERPLANK, 2000), (II; NOWELL, 2000) e (FORLIZZI; FORD, 2000), sendo os mesmos sobre qualidade de experiência física em projeto de tecnologias embarcadas, ambientes virtuais e projeto de experiência de usuário - com foco em projeto - respectivamente.

Em 2001 são encontradas 3 referências: (HONDA; NINOMIYA; SASSA, 2001), (MORRIS; TURNER, 2001) e (WOLTER; MOORSEL, 2001). A primeira é um relatório técnico não relacionado sobre sistemas de qualidade de cabines de ar; o segundo já uma referência aceitável também encontrada na pesquisa no *Web of Knoledge*; e o terceiro o trabalho técnico de Wolter e Van Moorsel “*The relationship between quality of service and business metrics: Monitoring, notification and optimization*”, também com uma referência relevante.

Com os dados coletados nestas duas ferramentas de pesquisa pode-se então concluir que o termo começou a ser utilizado com a conotação esperada por volta de 2001, com utilização relevante do uso do termo encontrado nas duas ferramentas neste ano. Mesmo com o termo sendo aplicado em algumas áreas com conotação fora da esperada, em alguns casos particulares, a data inicial parece convergir para este ano.

Um fato importante a ser percebido é que o comportamento nor-

Scopus Databases					
Article Title, Abstract and Keywords: ("Quality of Experience" or QoE)					
Ano	Geral	Ciências Físicas	Ciência da Computação	Engenharia	Ciênc. da Comp. ou Engenharia
1960-2014	2390	2209	1794	912	2177
2014	2	2	1	2	2
2013	440	428	352	173	421
2012	587	571	473	243	566
2011	416	407	351	144	404
2010	332	312	260	108	309
2009	226	222	188	103	220
2008	131	121	89	67	120
2007	77	68	42	32	66
2006	45	39	22	21	38
2005	26	16	8	7	14
2004	15	6	2	5	6
2003	9	3	2	2	3
2002	9	1	0	0	0
2001	10	4	2	1	3
2000	10	3	1	2	3
1999	3	0	0	0	0
1998	9	1	1	1	1
1990-1997	22	3	0	0	0
1989	2	1	0	1	1
1954-1988	19	1	0	0	0

Tabela 6: Pesquisa no *Scopus*. Título de artigo, resumo e palavras chave selecionadas: ("Quality of Experience" OR QoE) em <http://www.scopus.com> realizada em 8 de Janeiro de 2014.

mal estatístico foi encontrado, como esperado, com a ferramenta *Scopus*, como pode ser visto na figura 2.

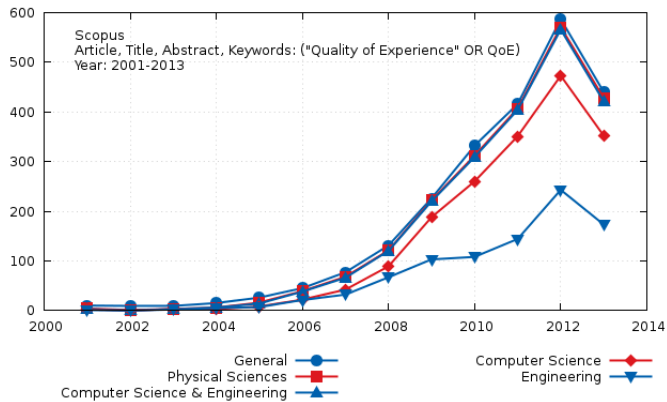


Figura 2: Histograma de comportamento de pesquisa em *Scopus*. Título de artigo, resumo, palavras chave selecionado: ("Quality of Experience" OR QoE) de 2001 a 2013 em <http://apps.webofknowledge.com> obtido em 8 de Janeiro de 2014.

Com os resultados desta ferramenta foi percebido que não apenas *ciência da computação* teve mais resultados do que *engenharia*, e que as duas áreas de pesquisa combinadas continham quase todos os artigos encontrados, mas também que o volume de dados obtido com esta ferramenta foi maior porém muito próximo ao volume das bases do *Web of Knowledge* até 2009. O que comprova a possibilidade levantada de negligência quanto a publicações indexadas em *ciência da computação* nestas primeiras bases utilizadas.

A tabela 7 mostra os resultados da pesquisa (“Quality of Experience” OR QoE) AND (Database OR “Distributed Architecture” OR “Big Data”), com 64 publicações nas áreas combinadas de *ciência da computação* e *engenharia*.

Scopus Databases					
Article Title, Abstract and Keywords: (“Quality of Experience” or QoE) AND (Database OR “Distributed Architecture” OR “Big Data”)					
Ano	Geral	Ciências Físicas	Ciência da Computação	Engenharia	Ciêc. da Comp. ou Engenharia
1960-2014	68	64	62	20	64
2014	0	0	0	0	0
2013	23	21	21	8	21
2012	19	19	18	6	19
2011	9	9	9	2	9
2010	9	8	8	2	8
2009	2	2	1	1	2
2008	2	2	2	1	2
2007	2	2	2	0	2
2002-2006	0	0	0	0	0
2001	1	1	1	0	1
1955-2000	0	0	0	0	0
1954	1	0	0	0	0

Tabela 7: Pesquisa no *Scopus*. Título de artigo, resumo, palavras chave selecionado: (“Quality of Experience” OR QoE) AND (Database OR “Distributed Architecture” OR “Big Data”) em <http://www.scopus.com> realizada em 12 de Fevereiro de 2014.

Destes 64 artigos, apresentados nas tabelas 8, 9 e 10, o primeiro de 2001 em *ciência da computação* de (WOLTER; MOORSEL, 2001), como previamente mencionado, “lida com os efeitos da degradação de qualidade de serviço no lucro de e-services” (WOLTER; MOORSEL, 2001) mas não é focado em *BDDs*, e os outros resultados são em geral relacionados com redes e *peer to peer* (como os artigos 02, 03, 05, 16, 20, 30, 37, 39, 41, 50, 58 e 61), video, VoIP e sistemas de fala (como os artigos 07, 08, 11, 12, 17, 21, 22, 23, 24, 25, 28, 38, 40, 42, 46, 47, 55, 60 e 62), e tecnologias estereoscópicas de vídeo e imagem (como os artigos 01, 06, 14, 18, 19, 27, 35, 43, 51, 52, 53, 54 e 57), da mesma maneira que os resultados obtidos do *Web of Knowledge*. Alguns resultados (04, 29, 33, 34, 48 e 63) são referências a anais de congressos e não a publicações

específicas, e então estes resultados foram considerados como *erros* nos resultados da pesquisa.

Os artigos relevantes encontrados, com foco em *BDDs*, foram os mesmos dois artigos de Rogério Luís de Carvalho Costa e Pedro Furtado (COSTA; FURTADO, 2011, 2013), mencionados anteriormente como resultados obtidos com o *Web of Science* na tabela 5 e detalhados na seção 3.5: *Detalhes da revisão bibliográfica*.

Os resultados sublinhados nas tabelas 8, 9 e 10 foram encontrados nas bases do *Scopus* e *Web of Knowledge* simultaneamente.

Scopus Databases		
n	Artigo	Citações
01	"DCT-based objective quality assessment metric of 2D/3D image" (LIU; SUN; YANG, 2013)	0
02	"SmartenIT cloud traffic management approach and architectural considerations" (PAPAFILI et al., 2013)	0
03	"Logging real packet reception patterns for end-to-end quality of experience assessment in wireless multimedia transmission" (SLADOJEVIC et al., 2013)	0
04	"10th Int. Joint Conf. on ICETE 2013; 4th Int. Conf. on DCNET 2013, 10th Int. Conf. on ICE-B 2013 and OPTICS 2013 - 4th Int. Conf. on Optical Communication Systems, Proc." (ICETE, 2013)	0
05	"A feasible solution to provide cloud computing over optical networks" (TAHERI; ANSARI, 2013)	0
06	"Paired comparison-based subjective quality assessment of stereoscopic images" (LEE; GOLDMANN; EBRAHIMI, 2013)	2
07	"A dynamic system model of time-varying subjective quality of video streams over HTTP" (CHEN et al., 2013)	0
08	"Robustness of speech quality metrics to background noise and network degradations: Comparing ViSQOL, PESQ and POLQA" (HINES et al., 2013)	0
09	"Use- and QoE-related aspects of personal cloud applications: An exploratory survey" (VANDENBROUCKE; MOOR; MAREZ, 2013)	0
10	"How much longer to go? The influence of waiting time and progress indicators on quality of experience for mobile visual search applied to print media" (CAO; RITZ; RAAD, 2013)	0
11	"High definition H.264/AVC subjective video database for evaluating the influence of slice losses on quality perception" (STAELENS et al., 2013)	0
12	"Perceptual experience of time-varying video quality" (REHMAN; WANG, 2013)	0
13	"Providing quality of experience for users: The next DBMS challenge" (COSTA; FURTADO, 2013)	0
14	"Subjective evaluation of stereoscopic image quality" (MOORTHY et al., 2013)	3
15	"A network-aware virtual machine placement algorithm in mobile cloud computing environment" (CHANG et al., 2013)	0
16	"Optimal design of virtual networks for resilient cloud services" (BARLA et al., 2013)	0
17	"Kodlama tabanlı bozulmalar üzerine uzam- zamansal referanssız video kalite degerlendirme modeli [Spatiotemporal no-reference video quality assessment model on distortions based on encoding]" (ZERMAN et al., 2013b)	0
18	"Towards standardized 3DTV QoE assessment: Cross-lab study on display technology and viewing environment parameters" (BARKOWSKY et al., 2013)	0
19	"A survey on 3D quality of experience and 3D quality assessment" (MOORTHY; BOVIK, 2013)	0
20	"Approach for service search and peer selection in P2P service overlays" (FIORESE; SIMOES; BOAVIDA, 2013)	0

Tabela 8: Pesquisa no *Scopus*. Título de artigo, resumo, palavras chave selecionadas: ("Quality of Experience" OR QoE) AND (Database OR "Distributed Architecture" OR "Big Data") resultados de 2013 em *ciência da computação* e *engenharia* em <http://www.scopus.com> obtidos em 12 de Fevereiro de 2014.

Scopus Databases		
n	Artigo	Citações
21	"On quality of experience of scalable video adaptation" (LI; CHEN; TAN, 2013)	1
22	"QoE analysis for scalable video adaptation" (LI; CHEN; TAN, 2012a)	0
23	"QoE-aware resource allocation for scalable video transmission over multiuser MIMO-OFDM systems" (LI; CHEN; TAN, 2012b)	0
24	"PNN-based QoE measuring model for video applications over LTE system" (HE et al., 2012)	0
25	"QoE assessment of multimedia video consumption on tablet devices" (FLORIS et al., 2012)	0
26	"A reputation based vertical handover decision making framework (R-VHDF)" (LOUKIL et al., 2012)	0
27	"Comparison of stereoscopic technologies in various configurations" (FLIEGEL et al., 2012)	0
28	"Comparison of objective quality metrics on the scalable extension of H.264/AVC" (LEE, 2012)	0
29	"Proceedings - 2012 3rd IEEE International Conference on Network Infrastructure and Digital Content, IC-NIDC 2012" (IC-NIDC, 2012)	0
30	"PP2db: A privacy-preserving, P2P-based scalable storage system for mobile networks" (CROTTI et al., 2011)	0
31	"Empirical study based on machine learning approach to assess the QoS/QoE correlation" (MUSHTAQ; AUGUSTIN; MELLOUK, 2012)	0
32	"Supporting wireless access markets with a user-centric QoE-based geo-database" (FORTETSANAKIS et al., 2012)	0
33	"MobiArch'12 - 7th ACM International workshop on mobility in the evolving internet architecture" (MOBIARCH'12, 2012)	0
34	"2012 4th International Workshop on Quality of Multimedia Experience, QoMEX 2012" (QOMEX, 2012)	0
35	"Quality of experience assessment for stereoscopic images" (QI et al., 2012)	1
36	"Rendering 3-D high dynamic range images: Subjective evaluation of tone-mapping methods and preferred 3-D image attributes" (MAI et al., 2012)	0
37	"Peer selection in P2P service overlays using geographical location criteria" (FIORESE; SIMOES; BOAVIDA, 2012)	0
38	"Guidelines for an improved quality of experience in 3-D TV and 3-D mobile displays" (XU; CORIA; NASIOPOULOS, 2012)	0
39	"Resilient virtual network design for end-to-end cloud services" (BARLA; SCHUPKE; CARLE, 2012)	0
40	"QoE prediction model and its application in video quality adaptation over UMTS networks" (KHAN; SUN; IFEACHOR, 2012)	15
41	"QoE content distribution network for cloud architecture" (TRAN; MELLOUK; HOCEINI, 2011)	0
42	"A flexible QoE framework for video streaming services" (ALVAREZ et al., 2011)	1
43	"Objective metrics for quality of experience in stereoscopic images" (KING et al., 2011)	1
44	"An approach to peer selection in service overlays" (FIORESE; SIMES; BOAVIDA, 2011)	0
45	"Quality of experience in distributed databases" (COSTA; FURTADO, 2011)	1

Tabela 9: Pesquisa no *Scopus*. Título de artigo, resumo, palavra chave selecionado: ("Quality of Experience" OR QoE) AND (Database OR "Distributed Architecture" OR "Big Data") resultados de 2012 em *ciência da computação e engenharia* em <http://www.scopus.com> obtidos em 12 de Fevereiro de 2014.

Scopus Databases		
n	Artigo	Citações
46	"Evaluation of video quality metrics on transmission distortions in H.264 coded video" (SEDANO et al., 2011)	0
47	"A comparison study of automatic speech quality assessors sensitive to packet loss burstiness" (JELASSI; RUBINO, 2011a)	3
48	"Proceedings of SPIE-IS and T Electronic Imaging - Multimedia on Mobile Devices 2011; and Multimedia Content Access: Algorithms and Systems V" (SPIE-IS, 2011)	0
49	"Visual quality assessment algorithms: What does the future hold?" (MOORTHY; BOVIK, 2011)	19
50	"Analysis of overlay topology of peer-to-peer applications" (YU et al., 2010)	1
51	"Estimating quality of experience on stereoscopic images" (XING et al., 2010a)	0
52	"Temporal synchronization in stereoscopic video: Influence on quality of experience and automatic asynchrony detection" (GOLDMANN; LEE; EBRAHIMI, 2010)	3
53	"An objective metric for assessing quality of experience on stereoscopic images" (XING et al., 2010b)	1
54	"Guidelines for capturing high quality stereoscopic content based on a systematic subjective evaluation" (XU; CORIA; NASIOPOULOS, 2010)	0
55	"A user-perceived video quality assessment metric using inter-frame redundancy" (SHI; DING; LI, 2010)	6
56	"Proposed framework for evaluating quality of experience in a mobile, testbed-oriented living lab setting" (MOOR et al., 2010)	23
57	"A comprehensive database and subjective evaluation methodology for quality of experience in stereoscopic video" (GOLDMANN; SIMONE; EBRAHIMI, 2010)	17
58	"Foreword-cognitive radio: From equipment to networks" (MOY; DOYLE; SANADA, 2009)	0
59	"Qualia: The geometry of integrated information" (BALDUZZI; TONONI, 2009)	33
60	"New single-ended objective measure for non-intrusive speech quality evaluation" (MAHDI; PICOVICI, 2010)	0
61	"Optimizing user QoE through overlay routing, bandwidth management and dynamic transcoding" (WIJNANTS et al., 2008)	0
62	"QoE monitoring platform for video delivery networks" (VERA; RODRIGUEZ-BOCCA; RUBINO, 2007)	1
63	"Proceedings of the 7th Nordic Signal Processing Symposium, NORSIG 2006" (NORSIG, 2006)	0
64	"The relationship between quality of service and business metrics: Monitoring, notification and optimization" (WOLTER; MOORSEL, 2001)	0

Tabela 10: Pesquisa no *Scopus*. Título de artigo, resumo, palavras chave selecionado: ("Quality of Experience" OR QoE) AND (Database OR "Distributed Architecture" OR "Big Data") resultados de 2001-2011 em *ciência da computação e engenharia* em <http://www.scopus.com> obtidos em 12 de fevereiro de 2014.

3.4.3 Análise estatística de interesse de pesquisa

Inicialmente, pela análise realizada na base *Web of Knowledge*, foi percebido que o assunto parecia ter tido pico de interesse em 2009 e novamente em 2012, como mencionado previamente. Entretanto, ao se analisar a base *Scopus*, percebeu-se uma distribuição normal bem comportada com pico em 2012, como demonstrado na figura 3 na área de pesquisa de *ciência da computação*, e na figura 4 quando se considerando combinadas as áreas de *ciência da computação* e *engenharia*.

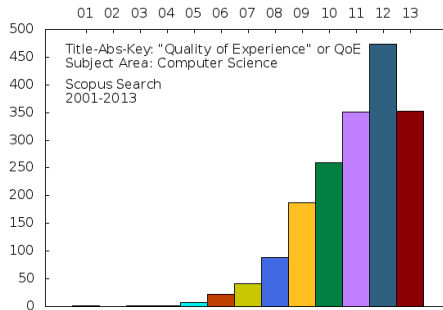


Figura 3: Distribuição normal bem comportada da base *Scopus* na área de pesquisa de *ciência da computação*. Resultados para o tópico: “Quality of Experience” or QoE de 2001 a 2013 obtidos em 8 de Janeiro de 2014.

Quando se considera as áreas de pesquisa de *ciência da computação* e *engenharia* combinadas, também parece existir um pico de interesse em 2012 na base *Web of Knowledge*, como demonstrado na figura 6, diferente dos resultados sobre *ciência da computação* apresentados na figura 5. Entretanto, os dados obtidos parecem levar a um aspecto mais complexo a ser considerado. O comportamento dessas bases parece se distanciar depois de 2009. Antes desta data, o número de resultados parece estar em uma consistência aceitável, não apenas comportamental mas também em termos de magnitude.

Observando-se mais cuidadosamente é possível se notar que até mesmo antes de 2009 a base *Web of Knowledge* já lidava com um volume menor de dados que a base *Scopus*. Entretanto, parece que especialmente depois de 2009 as fontes do *Web of Knowledge* não cresceram como as do *Scopus*, e é por isso que este efeito é percebido, como se o *Web of Knowledge* tivesse negligenciado as fontes de *ciência da com-*

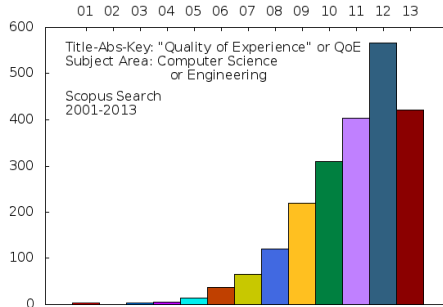


Figura 4: Distribuição normal bem comportada da base *Scopus* nas áreas de pesquisa de *ciência da computação* e *engenharia* combinadas. Resultados para o tópico: “Quality of Experience” or QoE de 2001 a 2013 obtidos em 8 de Janeiro de 2014.

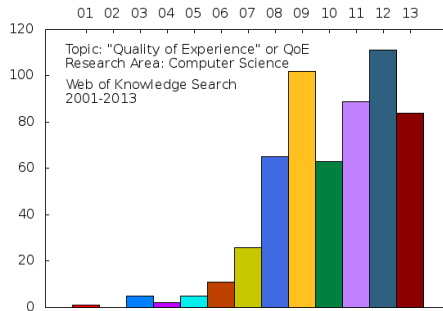


Figura 5: Distribuição normal *não bem comportada* da base *Web of Knowledge* na área de pesquisa de *ciência da computação*. Resultados para o tópico: “Quality of Experience” or QoE de 2001 a 2013 obtidos em 6 de Janeiro de 2014.

putação, nas quais este estudo está precisamente interessado em.

Pode ser então assumido que o pico de uso do termo em pesquisas foi em 2012 com o comportamento estatístico normal esperado.

E, mesmo com o pico de interesse já tendo sido alcançado, a existência de estudos sobre *QoE*, ou mesmo *QoS* de uma maneira que possa ser aplicada a *considerações QoE* em *BDDs*, é quase inexistente.

A ausência de estudos sobre consideração de serviços a serem avaliados, em um modelo servidor como em *BDDs*, parece um vazio

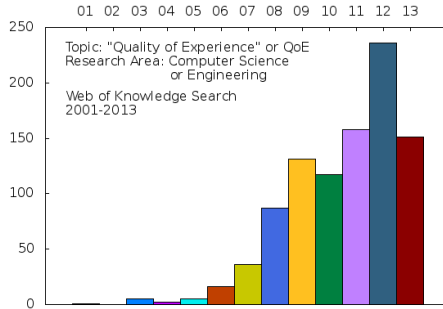


Figura 6: Distribuição normal *não bem comportada* da base *Web of Knowledge* nas áreas de pesquisa de *ciência da computação* e *engenharia* combinadas. Resultados para o tópico: “Quality of Experience” or QoE de 2001 a 2013 obtidos em 6 de Janeiro de 2014.

que precisa ser preenchido com os conceitos já existentes na literatura. Sendo esta ausência um grande motivador para realização desta tese.

3.5 DETALHES DA REVISÃO BIBLIOGRÁFICA

Esta breve revisão é focada nos tópicos combinados de *arquiteturas de BDDs* e *QoE* de maneira a não considerar os assuntos de maneira independente.

Mesmo os conceitos de ambos os tópicos sendo interessantes, os resultados de pesquisas em separado são em geral associados a campos de pesquisa fora do interesse do presente estudo.

3.5.1 Artigo: Quality of experience in distributed databases

A publicação de 2001 de Rogério Luís de Carvalho Costa e Pedro Furtado, “*Quality of experience in distributed databases*” (COSTA; FURTADO, 2011), alega ser a primeira introdução ao uso de *QoE* aplicada a *BDDs*. Ela é uma proposta para a melhoria do nível de *QoE* que pode ser assumidamente provista por *sistemas de BDDs* com o foco principal em mecanismos que podem aumentar a *satisfação do usuário* na questão de acesso a estes *sistemas de BDDs*.

Vale considerar o ponto de que este estudo lida com o pressuposto de que “em sistemas de bases de dados tradicionais, usuários

não podem especificar restrições relacionadas a execução”, concluindo que “o sistema de bases de dados não tem como avaliar se as expectativas do usuário são satisfeitas e nem pode o sistema tomar ações corretivas quando necessário” (COSTA; FURTADO, 2011). É um ponto extremamente válido, mas com o pressuposto de que o usuário precisa ter conhecimento específico de operações em bases de dados, bem como liberdade de decisão neste nível, para poder lidar com restrições de execução. Fazendo mais sentido, de maneira mais abstrata, o desenvolvimento de sistemas com esta camada abaixo, em vez de se esperar que o usuário lide com isso propriamente.

Isto é exatamente o que (COSTA; FURTADO, 2011) reivindica: “um sistema de base de dados orientado a QoE”, agrupando definições de *QoS* que permitem o uso de *QoE*, e não uma *solução QoE* propriamente dita. Esta é uma primeira *introdução* a estes conceitos aplicados a *sistemas de BDDs*, e uma primeira abertura que permite que definições de *QoS* sejam usadas para o desenvolvimento de *sistemas de QoE* de maneira que estes conceitos possam ser aplicados a *sistemas de BDDs*, com modelos teóricos de *sistemas de QoE* podendo ser então aplicados a *BDs*.

No *modelo QoE-DDB* apresentado, “cada comando de usuário é traduzido em tarefas que são executadas por serviços de dados chamados *módulos comunitários*, e *serviços de dados locais* negociam *objetivos em nível de serviço (ONS)* para cada tarefa, melhorando a confiança do sistema” (COSTA; FURTADO, 2011). São apresentados também “programação orientada a *QoE* e estratégias de colocação dinâmica de dados” (COSTA; FURTADO, 2011).

Eles propõem sete tipos de *requisições de acesso a dados (RADs)*, que são “expressas em termos de restrições de execução de consultas” (COSTA; FURTADO, 2011), com as seguintes definições:

- **Exigência de disponibilidade de dados:** indicando períodos em que um certo dado deve estar disponível;
- **Exigência de atualização de dados:** indicando o quão atualizada uma réplica de um certo dado deve estar para poder ser utilizada como resposta a uma pesquisa;
- **Exigência de prazo de execução:** especificando o intervalo em que a execução do comando deve ser finalizada;
- **Exigência de modo de execução desconectado:** indicando que o comando deve ser executado mesmo se o usuário for desconectado do sistema;

- **Exigência de prioridade de execução:** especificando a prioridade de execução do comando;
- **Exigência de tempo de início de execução:** especificando a data e hora em que a execução do comando deve iniciar;
- **Exigência de tempo de fim de execução:** especificando a data e hora em que o comando já deve ter sido finalizado.

Na sequência infere-se sobre a ausência de capacidade das “métricas de performance tradicionais, como tempo de resposta e taxa de transferência, para medir o *nível de QoE* que o sistema está provendo” (COSTA; FURTADO, 2011) para introduzir um conjunto de *indicadores de performance chave (IPCs)*. Estes *IPCs* são supostamente capazes de medir “a capacidade do sistema de alcançar as expectativas dos usuários” (COSTA; FURTADO, 2011).

Os *IPCs* propostos são:

- **Taxa de aceitação:** que provê a medida do número de comandos com *RADs* que o sistema concorda em executar (e satisfazer as *RADs* correspondentes);
- **Taxa de manutenção de compromisso:** que mede a capacidade do sistema em satisfazer as *RADs* que foram acordadas;
- **Taxa de sucesso:** que provê a medida da quantidade de comandos com *RADs* que o sistema executa enquanto satisfazendo *RADs* específicas;
- **Nível de QoE:** que provê a medida do nível de *QoE* que o sistema está provendo aos usuários.

Subsequentemente, como comentado acima, estes autores apresentam “programação *orientada a QoE* e estratégias de colocação dinâmica de dados” (COSTA; FURTADO, 2011). Este parece um modelo confiável que lida com um conjunto bem específico de indicadores de performance para medir o *nível de QoE* provido pelo sistema.

Eles também apresentam resultados experimentais em alguns cenários ressaltando a importância de:

- Avaliação de *RADs* antes da execução de consultas de usuários;
- Informar usuários sobre a possibilidade (ou não) de se satisfazer *RADs* especificadas;
- Não gastar poder computacional na execução de comandos em que as *RADs* não podem ser realizadas;

- E utilizar *RADs* para aumentar a satisfação dos usuários.

Finalizando com a conclusão de que a aplicação de *QoE* a *BDDs* pode realmente trazer ganhos de performance dentro das considerações do trabalho, com base nos experimentos apresentados. Infelizmente estes experimentos não são de fácil reprodução, lidando com uma base de simulação com código não disponibilizado pelos autores, e não podendo ser verificados de maneira alguma.

Os autores também deixam claro que os conceitos não foram implementados utilizando nenhuma base de dados real.

3.5.2 Artigo: Providing quality of experience for users: the next DBMS challenge

O artigo de 2013 de Rogério Luís de Carvalho Costa e Pedro Furtado, “*Providing quality of experience for users: the next DBMS challenge*” (COSTA; FURTADO, 2013), discute sobre a necessidade de “mecanismos orientados a *QoE*, atualmente não disponibilizados em sistemas de gerenciamento de bases de dados tradicionais” de maneira que estes possam lidar com a consideração de requisitos especificados pelo usuário e então, com base nisso, se possa ajustar as operações considerando tais expectativas.

O trabalho faz comparações de estratégias de gerenciamento de dados, reforçando o trabalho anterior e demonstrando o ganho que pode ser alcançado aplicando-se estratégias de *QoE* sobre *BDDs*.

Este é um trabalho com dados estatísticos referentes ao trabalho mencionado anteriormente, (COSTA; FURTADO, 2011). Assim sendo, não se vê relevância em se entrar em detalhes sobre este, já que são resultados sobre o artigo mencionado e não novos conceitos a serem considerados.

Porém, é digno de nota que ambos os trabalhos se limitam a consideração de que o usuário pode simplesmente definir o que espera, sem nenhuma restrição. Parecendo a abordagem literal de *QoS* de redes, mas com liberdade absoluta ao usuário. Ponto que é desconsiderado nestes trabalhos, afinal, tal ponto só faz sentido com um usuário utilizando uma base de dados local sem restrições, não sendo uma abordagem realmente útil para a proposição desta tese.

4 MODELAGEM ARQUITETURAL

O único modelo existente na literatura sobre *QoE* aplicada a *QoS* em *BDDs*, (COSTA; FURTADO, 2011), possui considerações de que usuários podem fazer exigências via comandos, e que comandos podem ser discriminados devido a tais exigências, não se adequando aos propósitos deste estudo. O trabalho subsequente, (COSTA; FURTADO, 2013), também indica que ainda não existem implementações, nem mesmo do modelo apresentado na publicação anterior, obrigando este estudo a definir um novo modelo. Parte-se então para a abordagem de definição de um modelo de mapeamento de *QoE* através de *QoS*, com as devidas definições de *QoS*, e definição arquitetural para aplicação sobre *BDDs*.

4.1 MODELO DE MAPEAMENTO TEÓRICO DE QUALIDADE DE EXPERIÊNCIA ATRAVÉS DE QUALIDADE DE SERVIÇO PARA BASES DE DADOS DISTRIBUÍDAS

O modelo de mapeamento teórico de *QoE* através de *QoS* aqui apresentado, como considerado a ser aplicado a *BDDs*, vem das definições de (BROOKS; HESTNES, 2010), onde eles afirmam que definições objetivas e quantitativas de *parâmetros de QoE* são um passo crucial. Assim sendo, e tentando evitar as considerações conceituais de *redes* dos requisitos hedônicos e estéticos do usuário (LAGHARI et al., 2012), já que em *BDDs* os *ANSs* parecem conectados a uma camada mais complexa que o usual melhor esforço, comum do ponto de vista de *redes*, um híbrido de *QoE* e *QoS* com considerações de intervalos de confiança é apresentado, mesmo sabendo-se não ser um modelo de *cobertura total de QoE*. Já que *QoE* é um conceito multidimensional centrado no usuário, é difícil para um sistema dar garantias exatas deste tipo.

Como *BDDs* podem ser usualmente enxergadas como arquiteturas de *provedor de serviços*, é fácil se definir valores exatos a serem providos, e já que os serviços são usualmente contratados com definições específicas, este modelo não fica tão abstrato quanto o modelo de um usuário qualquer consumindo um serviço genérico de internet. (SHENKER; PARTRIDGE; GUERIN, 1997) define que *métricas de QoS* devem ser garantias matematicamente prováveis, e aliando-se este conceito com a explicação de embasamento teórico da conexão dos *conceitos de QoE e QoS* em (LAGHARI et al., 2011), foca-se em dar garantias matemáticas de médias e desvios padrão para tentar alcançar a *QoE esperada*,

mesmo sabendo-se que isto é uma simplificação do conceito.

Os dois estudos previamente apresentados, (COSTA; FURTADO, 2011, 2013), mostram resultados práticos da aplicação de *métricas de QoS* - neste ponto a chamada *QoE* parece estritamente conectada com métricas de avaliação do nível acordado de *QoS* - para *BDDs* e métodos associados com parâmetros para avaliar tais métricas, na verdade não mapeando parâmetros estritamente conectados à satisfação do usuário, como é definida na *conceitualização de QoE* em (LAGHARI et al., 2011). Estes dois artigos avaliam as medidas esperadas de *QoS* de acordo com um *ANS*, não sendo definições fortes, mas, ao invés disso, orientações para um sistema *consciente de QoE*. O chamado *nível de QoE* parece ser na verdade o nível de completude de *QoS* esperado como definido por um *ANS* - um novo conceito, a completude de *QoS* pode levar a conceitualização parametrizada de *QoE*.

Para esclarecer as pontuações sobre *ANSs*, não mencionadas nos trabalhos citados, na visão usual de *QoS* de um sistema, um usuário não tem como especificar o comportamento esperado, já que o comportamento esperado é conectado à *QoS como acordado*. Do ponto de vista de (COSTA; FURTADO, 2011, 2013) é assumido que o usuário pode especificar como espera que o comportamento se dê, o que somente é aceitável do ponto de vista de uma *camada de QoS controlada*. O que pode ser considerado no presente estudo, após a análise destes dois trabalhos, é que as definições devem supostamente ser utilizadas por uma camada que possa avaliar o comportamento esperado, e não o usuário propriamente dito, sendo esta a origem das considerações referentes à *ANS*.

Para definir um *modelo QoE-QoS*, o passo principal é definir, de maneira concisa, parâmetros de *QoE* que possam ser medidos em uma *abordagem de BDD*. E mesmo o artigo (BROOKS; HESTNES, 2010) pontuando que definições objetivas e quantitativas de *parâmetros de QoE* são um passo crucial, o mesmo não apresenta nenhum modelo específico com parâmetros de *QoE*, o que permite a conclusão de que *parâmetros de nível de QoE* são estritamente ligados às definições de *QoS* como acordadas, com garantias muito específicas para cada tipo particular de serviço. De maneira que, para ser possível se definir *QoE*, em termos de *mapeamento para QoS*, é crucial a definição de *parâmetros de QoS*. Em uma comparação com os conceitos de (ERL; PUTTINI; MAHMOOD, 2013) - veja o capítulo 2 -, (COSTA; FURTADO, 2011) apresenta as definições de *RADs* como segue:

- **Disponibilidade de dados:** que pode ser mensurada com a métrica de *taxa de disponibilidade* de (ERL; PUTTINI; MAHMOOD,

2013);

- **Atualidade de dados:** um requisito de execução de tarefa que não parece ser diretamente mapeado para métricas apresentadas em (ERL; PUTTINI; MAHMOOD, 2013) - não é uma parte da métrica de *taxa de confiabilidade*, sendo um novo conceito a ser considerado pelas definições deste estudo;
- **Tempo de início/fim de execução e prazo de entrega:** que engloba a quase totalidade das métricas de (ERL; PUTTINI; MAHMOOD, 2013), como a *métrica de tempo de início de instância*, que considera o tempo necessário para se iniciar uma instância, a *métrica de tempo de resposta* se a operação for síncrona, ou a *métrica de tempo de conclusão* se a tarefa for assíncrona, a *métrica de tempo médio para transição* como uma variável de tempo de recuperação para probabilidade de falha quando chaveando um serviço falho para uma instância replicada, a *métrica de tempo médio para recuperação de sistema* considerando-se a probabilidade de recuperação de serviço para uma falha completa, a *métrica de tempo médio entre falhas* para consideração da possibilidade de várias falhas no caso de uma tarefa muito longa, a *métrica de duração de interrupção* como uma métrica de probabilidade associada com a falha de todas as réplicas, a *métrica de capacidade de rede* para se saber se a operação é alcançável considerando-se as características de rede, a *métrica de capacidade de servidor* para se saber se os servidores provendo os serviços podem completar a tarefa no prazo de entrega, quando considerando-se serviços de armazenamento, a *métrica de capacidade de armazenamento de dispositivos* associada com a *métrica de armazenamento de escalabilidade horizontal*, e considerando variação de cargas de trabalho, as *métricas de escalabilidade de servidor horizontal e vertical*;
- **Modo de execução desconectado:** uma opção normalmente associada a tarefas de longa duração que não são diretamente associadas a necessidades temporais;
- **Prioridade de execução:** uma definição padrão de *QoS* como prioridade que, no caso de subtarefas de *RADs*, podem ser definições automatizadas associadas com grupos específicos predefinidos de *RADs*.

A informação mencionada acima sobre *nível de QoE*, classificada como o nível de *QoS* esperado a ser alcançado, e o nível de completude alcançado, não é a definição de *QoE* baseada em (LAGHARI et al., 2011),

mas pode ser considerada como tal por questão de simplificação. Adicionalmente, as definições difusas de *QoS* não devem ser tão difusas, como demonstrado na comparação de (COSTA; FURTADO, 2011) com o trabalho de (ERL; PUTTINI; MAHMOOD, 2013), considerando que métricas exatas de *QoS* são o passo crucial para as *definições de QoE*.

De acordo com estes três artigos (ERL; PUTTINI; MAHMOOD, 2013; COSTA; FURTADO, 2011, 2013) sobre redes distribuídas em nuvens e gestão e avaliação de *QoS* em *BDDs*, foi observado que a *capacidade R&Q (recursos e qualidade)*: com restrições sobre execução, armazenamento, redes e cargas de trabalho; e a *capacidade RAS¹ (confiabilidade/disponibilidade/facilidade de manutenção do inglês reliability/availability/serviceability)*: com restrições de confiabilidade, disponibilidade e facilidade de manutenção - incluindo restrições de tolerância a falhas -, podem ser definidas como as duas principais famílias de garantias de *QoS* quando se lidando com serviços em nuvens, e potencialmente com *BDDs*, como demonstrado nas tabelas 11 e 12, sendo estas novas proposições desta tese.

Estas métricas são quase todas baseadas nas métricas propostas em (ERL; PUTTINI; MAHMOOD, 2013), exceto pela *métrica de taxa de atualização das métricas de confiança*, que é baseada em necessidades de *BDDs*, como demonstrado em (COSTA; FURTADO, 2011), e com fundamentação em (WEI; SON; STANKOVIC, 2004), e a *métrica de replicação de armazenamento das capacidades de armazenamento*, que é uma definição padrão explícita para *BDs* baseadas em nuvem.

As garantias extras, como *modo de execução desconectado e prioridade de execução*, apresentadas em (COSTA; FURTADO, 2011), são consideradas inerentes ao processo de provimento de garantias de *QoS*, e, por esta razão, não são aqui apresentadas como métricas diretamente mensuráveis.

Agora, com *parâmetros de QoS* bem definidos e com métricas mensuráveis bem especificadas, a definição de parâmetros de *QoE* pode ser delineada e mapeada para as referidas *métricas de QoS*.

A identificação de *métricas de QoE* baseadas em expectativas de usuários, de acordo com (SOLDANI; LI; CUNY, 2006), pode ser genericamente agrupada “sob duas categorias: *confiabilidade* e *qualidade*”. Isto pode parecer inicialmente um *excesso de simplificação*, mas *confiabilidade* pode ser melhor substituída pela definição de *RAS*, que é um conceito mais compreensível, e *qualidade*, que parece ser estritamente dirigida a considerações para sistemas celulares do artigo anteriormente

¹RAS: Um termo originalmente utilizado pela IBM para descrever a robustez de seus computadores *mainframe* (SIEWIOREK; SWARZ, 1998).

Tipos de capacidade	Métrica aplicável
Capacidade de execução	<u>Métrica de capacidade de servidor</u> → CPU: número → Frequência de CPU: GHz → Tamanho de RAM: GBs → Tamanho de armazenamento: GBs → Replicação: número de réplicas <u>Métrica de tempo médio de iniciação de instância</u> → Tempo iniciado - tempo requerido: média em minutos <u>Métrica de desvio padrão de inicialização de instância</u> → Desvio padrão de (Tempo iniciado - tempo requerido) <u>Métrica de tempo de resposta médio (operação síncrona)</u> → Tempo médio de resposta: média em milissegundos <u>Métrica desvio padrão de tempo de resposta (operação síncrona)</u> → Desvio padrão de envio de resposta: média em milissegundos <u>Métrica de tempo médio de realização (tarefa assíncrona)</u> → Tempo médio de resposta: média em segundos <u>Métrica de desvio padrão de tempo de realização (tarefa assíncrona)</u> → Desvio padrão do tempo de resposta: média em milissegundos
Capacidade de armazenamento	<u>Métrica de capacidade de armazenamento de dispositivo</u> → Capacidade de armazenamento de dispositivo: GBs <u>Métrica de escalabilidade de armazenamento horizontal</u> → Alterações de armazenamento permissíveis ao aumento de carga de trabalho: GBs <u>Métrica de replicação de armazenamento</u> → Número de réplicas: Número mínimo garantido
Capacidade de rede	<u>Métrica de capacidade de rede</u> → Largura de banda, taxa de transferência: MBs/s → Atraso, variação de atraso: milissegundos <u>Métrica de capacidade de aplicação</u> → Capacidade de aplicação: requisições/min
Capacidade de carga de trabalho	<u>Métrica de escalabilidade horizontal de servidor</u> → Alterações admissíveis a aumento de carga de trabalho: número de servidores virtuais no <i>pool</i> de recursos <u>Métrica de escalabilidade vertical de servidor</u> → Capacidade admissível de flutuações de carga de trabalho: número flutuante de CPUs e tamanho de RAM em GBs

Tabela 11: Garantias de QoS do tipo: Capacidades R&Q, como propostas na tese.

mentionado, ficando melhor substituída pela definição de REQ .

Em outra abordagem, (GONG et al., 2009) apresenta um modelo muito consistente para medição de QoE , chamado de modelo pentagrama, que é composto pelos indicadores de performance chave: integralidade, capacidade de retenção, disponibilidade, usabilidade e instantaneidade. Este artigo, como (SOLDANI; LI; CUNY, 2006), lida certamente com tecnologias de *redes* não diretamente aplicáveis ao paradigma de *BDDs*. Mas este segundo artigo, (GONG et al., 2009), interessante lida com fórmulas exatas de avaliação, e parece ser uma boa orientação ao modelo como pretendemos definir.

Os “fatores que influenciam QoE e suas mais importantes medições”, como considerado em (GONG et al., 2009), são apresentados em um modelo mais generalizado na tabela 13.

Com base nestes indicadores de performance chave apresentados na tabela 13, o artigo (GONG et al., 2009) considera a fórmula de ava-

Tipos de capacidade	Métrica aplicável
Capacidade de confiabilidade	<u>Métrica de taxa de sucesso</u> → Resultados de serviços bem sucedidos: número de respostas bem sucedidas em porcentagem <u>Métrica de taxa de atualização</u> → Taxa de atualização de acesso (TAA): $\frac{FrequenciaDeAcesso}{FrequenciaDeUpdate}$
Capacidade de disponibilidade	<u>Métrica de taxa de tempo em operação</u> → Porcentagem de tempo de operação de serviço: $\frac{TempoTotalEmOperacao}{TempoTotal}$
Capacidade de facilidade de manutenção	<u>Métrica de duração de interrupção</u> → Duração de uma única interrupção: tempo de fim de interrupção - tempo de início de interrupção <u>Métrica de tempo médio entre falhas</u> → Tempo entre falhas consecutivas de um serviço: $\frac{DuracaoDePeriodoNormaldeOperacao}{NumeroDeFalhas}$ <u>Métrica de tempo médio de transição</u> → Tempo de transição a partir de uma falha: minutos <u>Métrica de tempo médio de recuperação de sistema</u> → Tempo para uma recuperação completa de uma falha de serviço: minutos

Tabela 12: Garantias de QoS do tipo: Capacidades RAS, como propostas na tese.

liação de QoE , com os valores de a , b , c , d e e entre zero e um, com a área do pentagrama variando de zero a ~ 2.37775 - considerando-se $\sin(72^\circ) \cong 0.9511$ -, como demonstrado na equação 4.1.

$$\begin{aligned}
QoE &= \frac{1}{2} \times \sin 72^\circ \times (ab + bc + cd + de + ea) \\
&\cong \frac{1}{2} \times 0.9511 \times (ab + bc + cd + de + ea) \\
&\cong 0.48 \times (ab + bc + cd + de + ea)
\end{aligned} \tag{4.1}$$

Indicadores de performance chave de QoE	Medidas mais importantes	Símbolo
Integralidade	Atraso, jitter e taxa de perdas	a
Capacidade de retenção	Taxa de interrupção	b
Disponibilidade	Taxa de sucesso de acesso	c
Usabilidade	Usabilidade	d
Instantaneidade	Tempo de resposta de conexão e acesso	e

Tabela 13: Fatores que influenciam QoE e suas medidas mais importantes (GONG et al., 2009).

Entretanto, um problema que é facilmente percebido neste modelo é que quando, a e c forem zero, o valor de b não tem influência sobre o resultado obtido. E, é claro, a avaliação com consideração de peso equalitário de parâmetros pode ser desenvolvida em um modelo

mais complexo, lidando com pesos diferenciados.

A solução para esta abordagem poderia ser a consideração de uma linha de cinco dimensões, com os valores de a , b , c , d e e definindo a outra extremidade de uma linha iniciada na origem $(0, 0, 0, 0, 0)$.

Além desta definição de quantificação, (KIM et al., 2008) apresenta um modelo de correlação *QoE-QoS*, para *avaliação de QoE* baseada em *parâmetros de QoS*, porém as métricas subjetivas de *QoE* sobre *opinião* não são viáveis para a abordagem como desejada neste estudo. Adicionalmente, os *parâmetros de QoS* utilizados, como em quase todos os estudos envolvendo este tópico, são a simples tríade de redes: atraso, variação de atraso e perda de informação.

Entretanto, (KIM et al., 2008) também pontua que, para cada serviço, “o *nível de QoS* requerido para satisfazer uma *classe QoE* é diferente”. E, como o objetivo do presente estudo é o de se encontrar definições para *BDDs*, podemos definir os *serviços* como: inserção, atualização, deleção e seleção.

Neste ponto, tendo-se as *métricas de QoS* bem definidas e os serviços a serem avaliados também especificados, o passo de conclusão é o de se definir *métricas de QoE* associadas com cada serviço mapeadas em termos de *métricas de QoS*.

É digno de nota que a medição da *completude de QoS* na avaliação de *nível de QoE*, considerando-se um *acordo bem definido de QoS*, não é uma medição direta, mas ainda assim é crucial para avaliação quando considerando se evitar o questionamento habitual sobre como o usuário se sente sobre a experiência do serviço consumido, sendo este um fator de influência direta na maneira como o usuário irá se sentir sobre esta experiência.

4.1.1 Serviço de inserção: garantias de QoS e medição de QoE

Considerando-se o serviço de inserção, a principal preocupação é facilmente inferida como a *capacidade de disponibilidade*, quando se considera as *garantias de QoS* a serem avaliadas de maneira preventiva, tal como para capacidade que influencia diretamente a *capacidade de confiabilidade*, a principal preocupação é a *avaliação de QoE* do ponto de vista do usuário. A *métrica de taxa de atualização* não tem influência nas garantias de serviço de inserção, afinal o quão novo é um dado não é uma métrica aplicável ao se adicionar novos dados.

A *capacidade de disponibilidade* é diretamente conectada com a *capacidade de manutenção* e, uma vez medida, a *manutenção* já está

considerada nas garantias do serviço provisto, mas também deve ser verificada para correções quanto a comportamentos inesperados e *recuperação de QoS*.

Garantias de QoS do serviço de inserção		
Garantia	Tipo de capacidade	Métrica
R&Q	Capacidade de Execução	Tempo médio de inicialização de instância Desv. pad. de tempo de inicialização de instância Tempo médio de resposta Desv. Pad. de tempo de resposta
RAS	Capacidade de confiabilidade Capacidade de disponibilidade	Taxa de sucesso Taxa de tempo em operação

Tabela 14: Garantias de QoS do serviço de inserção.

É importante notar que medir direta ou indiretamente a *capacidade de manutenção* não é uma maneira de se medir garantias de manutenção, mas sim de possibilitar a correção de comportamento do sistema de maneira a se atingir o *nível de QoS* acordado. Garantir um tempo máximo entre falhas e medir o tempo médio entre falhas não é a mesma coisa. Mas, considerando o fato que este estudo visa lidar com a medição do *nível de QoE*, as definições de *QoS* podem ser *relaxadas* quando não afetando as *métricas de QoE*.

Com respeito ao efeito de garantias *R&Q* sobre as garantias *RAS*, elas são quase que garantias secundárias a estas, o que simplifica a necessidade de considerações nas fórmulas de avaliação *RAS*.

Levando-se em conta a *capacidade de confiabilidade* como a capacidade de maior influência sobre a experiência do usuário no tipo *RAS*, a mesma será considerada para *QoE* de usuário. Mas, como este estudo está também calculando a *completude de QoS*, como *QoE* do sistema, a *capacidade de disponibilidade* é também um fator de principal importância. É digno de nota que estas métricas não estão apenas inferindo a *QoE* de usuário, mas garantindo *QoE* em *nível de sistema*, já que a *capacidade de disponibilidade* é considerada junto com a *capacidade de confiabilidade*.

A equação 4.2 mostra a avaliação *RAS* de *QoS* em *nível de sistema* para o serviço de inserção de acordo com uma taxa acordada segundo um *ANS*.

$$QoE_{RAS}^{sistema}(ins) = \frac{TaxaDeUptime}{ANS(GarantiaRASInserção)} \quad [0; 1] \quad (4.2)$$

O valor é considerado como 1 quando o *taxa de uptime* do serviço é maior do que a garantia dada pelo *ANS*.

E, como discutido previamente, quando avaliando o nível de QoE como visto pelo usuário, o foco da avaliação RAS vai para a *taxa de sucesso* como demonstrado na equação 4.3.

$$QoE_{RAS}^{usuário}(ins) = \frac{TaxaDeSucesso}{ANS(GarantiaRASInserção)} \quad [0; 1] \quad (4.3)$$

O valor de $QoE_{RAS}^{usuário}(ins)$ é também considerado como 1 quando a *taxa de sucesso* do serviço é maior que a garantia dada pelo ANS .

As garantias de $R\&Q$ são basicamente referências temporais, checando se o tempo de inicialização de uma instância mais o tempo de execução está de acordo com o $ANS(GarantiaR\&QInserção)$ esperado. Isso dá um *cenário de pior caso* com garantia de $\sim 97.7\%$, considerando-se o comportamento normal com $\mu + 2\sigma$, com dois desvios padrão nos dois, tempo de inicialização de instância e tempo de resposta, quando considerando a avaliação em *nível de sistema* na equação 4.4, como a distribuição normal demonstrada na figura 7.

$$QoE_{R\&Q}^{sistema}(ins) = \frac{ANS(GarantiaR\&QInserção)}{((TempoMédioDeInicializaçãoDeInstância + 2 * DesvioPadrãoDeTempoDeInicialização) + (TempoMédioDeResposta + 2 * DesvioDePadrãoDeTempoDeResposta))} \quad [0; 1] \quad (4.4)$$

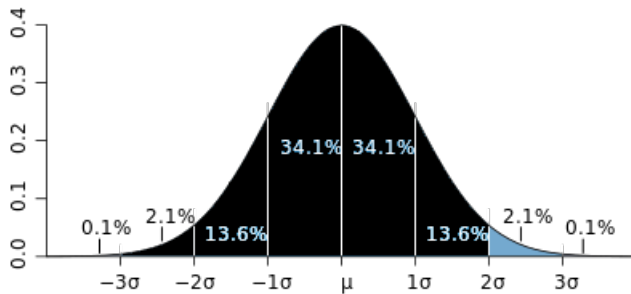


Figura 7: Desvio padrão acumulado para 2σ .

Assim sendo, baseado na avaliação em nível de sistema do nível de QoE , o sistema pode realizar tarefas de recuperação quando não correspondendo ao nível de QoE acordado.

Do ponto de vista do usuário, o tempo necessário para inicializar o serviço mais o tempo de resposta deve corresponder as garantias de tempo de $R\&Q$ dadas por um ANS , como demonstrado na equação 4.5.

$$QoE_{R\&Q}^{usuário}(ins) = \frac{ANS(GarantiaR\&QInserção)}{(TempoInicializaçãoInstância+TempoResposta)} \quad [0; 1] \quad (4.5)$$

Esta é mais uma avaliação pontual, transação a transação.

E então, de maneira a computar estes dois tipos de qualificação de QoE como uma única medida podemos simplesmente multiplicá-las, como demonstrado nas equações 4.6 e 4.7.

$$QoE_{total}^{sistema}(ins) = QoE_{RAS}^{sistema}(ins) * QoE_{R\&Q}^{sistema}(ins) \quad [0; 1] \quad (4.6)$$

$$QoE_{total}^{usuário}(ins) = QoE_{RAS}^{usuário}(ins) * QoE_{R\&Q}^{usuário}(ins) \quad [0; 1] \quad (4.7)$$

É interessante notar que a *formulação de QoE* será praticamente a mesma para todas as operações consideradas, mas os fatores que afetam os *tempos de resposta* e *taxa de sucesso* não serão os mesmos. Isso leva ao mesmo problema original de se lidar com considerações abstratas: medir o tempo médio e considerar o desvio padrão não são garantias, mas apenas medições que servem de ferramentas para se poder dar *garantias de QoS* e se lidar com recuperação quando lidando com comportamento inesperado. E então, neste ponto, é necessário não só se considerar a média e o desvio padrão como garantias de sistema a serem comparadas a um ANS , mas também se considerar que o sistema pode, de alguma maneira, tentar corrigir a *taxa de erros* inesperada.

4.1.2 Serviço de atualização: garantias de QoS e medição de QoE

O serviço de atualização é praticamente uma seleção parcial seguida de uma operação de inserção, e por estas definições irá funcionar praticamente como a inserção, mas com um pouco mais de atraso de resposta.

As equações e considerações de QoS são análogas às dadas na definição do serviço de inserção, como demonstrado nas equações 4.8, 4.9, 4.10, 4.11, 4.12 e 4.13.

Garantias de QoS do serviço de atualização		
Garantia	Tipo de capacidade	Métrica
R&Q	Capacidade de execução	Tempo médio de inicialização de instância Dev. pad. de tempo de inicialização de instância Tempo médio de resposta Desv. pad. de tempo de resposta
RAS	Capacidade de confiabilidade	Taxa de sucesso
---	Capacidade de disponibilidade	Taxa de tempo em operação

Tabela 15: Garantias de QoS do serviço de atualização.

$$QoE_{RAS}^{sistema}(atu) = \frac{TaxaDeUptime}{ANS(GarantiaRASAtualização)} \quad [0; 1] \quad (4.8)$$

$$QoE_{RAS}^{usuário}(atu) = \frac{TaxaDeSucesso}{ANS(GarantiaRASAtualização)} \quad [0; 1] \quad (4.9)$$

$$QoE_{R\&Q}^{sistema}(atu) = \frac{ANS(GarantiaR\&QAtualização)}{((TempoMédioDeInicializaçãoDeInstância + 2 * DesvioPadrãoDeTempoDeInicialização) + (TempoMédioDeResposta + 2 * DesvioPadrãoDeTempoDeResposta))} \quad [0; 1] \quad (4.10)$$

$$QoE_{R\&Q}^{usuário}(atu) = \frac{ANS(GarantiaR\&QAtualização)}{(TempoInicializaçãoInstância + TempoResposta)} \quad [0; 1] \quad (4.11)$$

$$QoE_{total}^{sistema}(atu) = QoE_{RAS}^{sistema}(atu) * QoE_{R\&Q}^{sistema}(atu) \quad [0; 1] \quad (4.12)$$

$$QoE_{total}^{usuário}(atu) = QoE_{RAS}^{usuário}(atu) * QoE_{R\&Q}^{usuário}(atu) \quad [0; 1] \quad (4.13)$$

4.1.3 Serviço de deleção: garantias de QoS e medição de QoE

As garantias do serviço de deleção são também análogas às considerações de inserção e atualização, exceto no *nível de QoS*, onde a capacidade de armazenamento não é relevante.

Garantias de QoS do serviço de deleção		
Garantia	Tipo de capacidade	Métrica
R&Q	Capacidade de execução	Tempo médio de inicialização de instância Desv. pad. do tempo de inicialização de instância Tempo médio de resposta Desv. pad. do tempo de resposta
RAS	Capacidade de confiabilidade Capacidade de disponibilidade	Taxa de sucesso Taxa de tempo em operação

Tabela 16: Garantias de QoS do serviço de deleção.

As equações 4.14, 4.15, 4.16, 4.17, 4.18 e 4.19 mostram as avaliações como consideradas.

$$QoE_{RAS}^{sistema}(del) = \frac{TaxaDeUptime}{ANS(GarantiaRASDeleção)} \quad [0; 1] \quad (4.14)$$

$$QoE_{RAS}^{usuário}(del) = \frac{TaxaDeSucesso}{ANS(GarantiaRASDeleção)} \quad [0; 1] \quad (4.15)$$

$$QoE_{R\&Q}^{sistema}(del) = \frac{ANS(GarantiaR\&QDeleção)}{((TempoMédioDeInicializaçãoDeInstância + 2 * DesvioPadrãoDeTempoDeInicialização) + (TempoMédioDeResposta + 2 * DesvioPadrãoDeTempoDeResposta))} \quad [0; 1] \quad (4.16)$$

$$QoE_{R\&Q}^{usuário}(del) = \frac{ANS(GarantiaR\&QDeleção)}{(TempoInicializacaoInstância+TempoResposta)} \quad [0; 1] \quad (4.17)$$

$$QoE_{total}^{sistema}(del) = QoE_{RAS}^{sistema}(del) * QoE_{R\&Q}^{sistema}(del) \quad [0; 1] \quad (4.18)$$

$$QoE_{total}^{usuário}(del) = QoE_{RAS}^{usuário}(del) * QoE_{R\&Q}^{usuário}(del) \quad [0; 1] \quad (4.19)$$

4.1.4 Serviço de seleção: garantias de QoS e medição de QoE

Em garantias de serviço de seleção, como nas garantias de deleção, a capacidade de armazenamento não é relevante. Aqui, a diferença influencional aparece no fato de a *taxa de atualização* ser também considerada, afinal quando se obtendo dados é importante se saber as garantias relativas a sua atualidade em um ambiente replicado.

Garantias de QoS para serviço de seleção		
Garantia	Tipo de capacidade	Métrica
R&Q	Capacidade de Execução	Tempo médio de inicialização de instância Desv. pad. do tempo de inicialização de instância Tempo médio de resposta Desv. pad. do tempo de resposta
RAS	Capacidade de confiabilidade	Taxa de sucesso Taxa de atualização
-----	Capacidade de disponibilidade	Taxa de tempo em operação

Tabela 17: Garantias de QoS para serviço de seleção.

Assim sendo, a *taxa de tempo de operação (uptime)* é utilizada em nível de sistema e a *taxa de sucesso* em nível de usuário, sendo multiplicadas pela *taxa de atualização* nas equações 4.20 e 4.21 para se adequarem ao comportamento esperado na avaliação de operações.

$$QoE_{RAS}^{sistema}(sel) = \frac{TaxaDeUptime * TaxaDeAtualização}{ANS(GarantiaRASSeleção)} \quad [0; 1] \quad (4.20)$$

$$QoE_{RAS}^{usuário}(sel) = \frac{TaxaDeSucesso * TaxaDeAtualização}{ANS(GarantiaRASSeleção)} \quad [0; 1] \quad (4.21)$$

As equações 4.22, 4.23, 4.24 e 4.25 são mantidas no mesmo formato das equações previamente apresentadas.

$$QoE_{R\&Q}^{Sistema}(sel) = \frac{ANS(GarantiaR\&QSeleção)}{((TempoMédioDeInicializaçãoDeInstância + 2 * DesvioPadrãoDeTempoDeInicialização) + (TempoMédioDeResposta + 2 * DesvioPadrãoDeTempoDeResposta))} \quad [0; 1] \quad (4.22)$$

$$QoE_{R\&Q}^{usuário}(sel) = \frac{ANS(GarantiaR\&QSeleção)}{(TempoInicializaçãoInstância + TempoResposta)} \quad [0; 1] \quad (4.23)$$

$$QoE_{total}^{sistema}(sel) = QoE_{RAS}^{sistema}(sel) * QoE_{R\&Q}^{sistema}(sel) \quad [0; 1] \quad (4.24)$$

$$QoE_{total}^{usuário}(sel) = QoE_{RAS}^{usuário}(sel) * QoE_{R\&Q}^{usuário}(sel) \quad [0; 1] \quad (4.25)$$

4.1.5 Efeito das equações sobre QoE do sistema e usuário

Nas equações 4.2, 4.8, 4.14 e 4.20 o $QoE_{RAS}^{sistema}(ins/atv/del/sel)$ é definido como completude da *taxa de tempo em operação* acordada, com o fator de *atualização* no caso especial de seleção. A *taxa de tempo em operação* é definida pela porcentagem do tempo de serviço em operação, e a *atualização* é definida como a taxa de atualização de acesso - veja tabela 12.

Desconsiderando-se os casos de média fora do intervalo de confiança acordado, quando o sistema verifica que a média está maior do que o valor esperado, ou o desvio padrão está fora das garantias acordadas, o sistema deve inicializar procedimentos de ajuste de desempenho de recuperação do sistema, de maneira a evitar que comportamento inesperado ocorra. O efeito pode ser de o sistema inicializar, ou chavear para, um espelho de serviço no *pool*, por exemplo.

Nas equações 4.3, 4.9, 4.15 e 4.21 o $QoE_{RAS}^{usuário}(ins/atv/del/sel)$ é definido como completude de *taxa de sucesso* acordada, também com o *fator de atualização* no caso especial de seleção. A *taxa de sucesso* é definida como os resultados de serviços bem sucedidos - veja tabela 12.

Esta é uma consideração que usualmente lida com um *pool* ativo

de provedores de serviço *backup* que irão automaticamente chavear de um servidor para outro de maneira transparente ao usuário consumindo o serviço. Se as garantias parecerem fora da média esperada, ou com um desvio padrão maior, mais serviços podem ser instantaneamente inicializados no *pool* de provedores de serviço.

O *tempo em operação* e a *taxa de sucesso* parecem bons parâmetros para avaliar a experiência do sistema e usuário, respectivamente, segundo o ponto de vista de *confiança*, *disponibilidade* e *facilidade de manutenção*. Por outro lado, do ponto de vista de *recursos* e *qualidade*, o tempo de resposta, lidando com o intervalo de confiança, parece uma melhor abordagem a ser seguida.

Nas equações 4.4, 4.10, 4.16 e 4.22 o $QoE_{R\&Q}^{sistema}(ins/atu/del/sel)$ é definido como completude do tempo esperado para que o serviço inicie e responda, com as considerações de desvio padrão já contidas na avaliação. Nos casos de média fora do intervalo esperado, ou desvios padrão acima, o sistema pode realizar ajustes de prioridade, provendo *mais recursos de hardware* aos serviços no *pool* para alcançar os tempos esperados.

Nas equações 4.5, 4.11, 4.17 e 4.23 o $QoE_{R\&Q}^{usuário}(ins/atu/del/sel)$ é definido como o tempo real de resposta contra o tempo esperado, um caso em que *mais recursos de hardware* podem ser reservados de maneira instantânea quando tentando-se corrigir o $QoE_{R\&Q}^{sistema}(ins/atu/del/sel)$ como um todo.

A combinação de equações, nomeadas equações 4.6, 4.12, 4.18 e 4.24, definindo $QoE_{total}^{sistema}(ins/atu/del/sel)$ e as equações 4.7, 4.13, 4.19 e 4.25, definindo $QoE_{total}^{usuário}(ins/atu/del/sel)$, são equações cruzando os tempos esperados com a *taxa de tempo em operação* e a *taxa de sucesso*, de maneira a simplificar as avaliações globais.

O efeito final da aplicação destas equações é o de dar ao sistema provedor maneiras de avaliar o comportamento atual e, com base no mesmo, prever comportamento futuro, de maneira a permitir correções antes que estados inesperados sejam alcançados. A correção destes comportamentos inesperados não está coberta neste modelo, já que é apenas um modelo de avaliação, mas será abordada com algumas considerações específicas nas seções seguintes.

Já que a *conceitualização de QoE* é holística, e *centrada no usuário*, a completude não é obviamente sujeita a revisão de maneira clara. Mas o modelo se propõe a tentar, com base em *definições de QoS*, a alcançar um melhor *nível de QoE* lidando com intervalos de confiança como especificados.

A viabilidade das considerações, bem como sua efetividade, são

avaliações mais aplicáveis a implementações específicas deste modelo genérico. Por exemplo, pode ser demonstrado que não é efetivo se disparar procedimentos de economia de energia quando a média está apenas um pouco menor que a esperada e com o desvio padrão esperado, já que a energia para se modificar o comportamento não irá evitar consumo de energia realmente, ou, em casos de erros extremos, que não é viável se recuperar em um determinado espaço de tempo, porque as definições de recuperação existentes simplesmente não cobrem o valor esperado, ou o sistema irá simplesmente *falhar*.

Por fim, a utilidade do modelo é clara, já que *serviços QoS* são serviços especificamente contratados. O modelo apenas propõe um ambiente que pode lidar com médias e desvios padrão para garantir aos usuários expectativa em um específico intervalo de confiança que provê os níveis esperados de *QoE*. É claro que esta visão de *QoE* como completude de uma *QoS* mais elaborada não é a *definição usual de QoE*, mas é suficiente para os propósitos deste trabalho.

4.1.6 Implicações do modelo de equações

Como mencionado na subseção anterior, o desvio padrão usado como complemento da média, de maneira a avaliar os serviços, é um parâmetro chave com confiabilidade notável para se alcançar um nível aceitável de *QoE*. Mesmo a *QoE* sendo um conjunto de parâmetros holísticos com o objetivo de tentar satisfazer os requisitos hedônicos e estéticos do usuário (LAGHARI et al., 2012), em uma abordagem *focada em sistemas*, uma *avaliação baseada em QoS* tem que o nível de conforto do usuário pode ser encapsulado na definição de intervalos de confiança.

Em uma exemplificação simples, se as *definições de QoS* garantem que uma inserção de dados de usuário em um formato específico, como o de redes sociais, irá ocorrer com uma média de n segundos, quando adiciona-se a garantia de n segundos com desvio padrão de d , o que é equivalente a $n \pm 2 * d$ em 95.4% dos casos ou $\leq n + 2 * d$ em 97.7% dos casos, esta garantia de *QoS* pode agora ser entendida mais como uma *garantia QoE-QoS híbrida*.

Assim sendo, agora é possível considerar *sistemas QoE-QoS* autocorrigíveis, onde as correções podem então ser aplicadas sob demanda a desvios padrão até que eles se encaixem nas necessidades do usuário.

Estas considerações de desvios padrão também apresentam a possibilidade para um sistema de tentativa de recuperação de um es-

tado inesperado de maneira mais lenta, sem a necessidade de se forçar o sistema a alcançar uma média esperada muito rapidamente. Desta maneira, o sistema não é forçado a extremos e o usuário tem a sensação de estabilização de serviços não abrupta.

O ponto principal é, entretanto, sobre recuperação do sistema. Um pequeno, ou ao menos controlado, desvio padrão é essencial para um sistema se recuperar corretamente. Não é sobre *boa recuperação*, ou recuperação de *qualidade média*, mas sobre recuperação em si. Um desvio padrão sem controle pode levar a replicação do estado que causou a falha inicial, levando a instabilidade em um chaveamento entre alto consumo de recursos e falha. Recuperar-se muito rápido a uma média esperada pode significar não se recuperar de maneira nenhuma.

Quando se considerando valores fora do intervalo de confiança, espera-se que o sistema tome medidas instantaneamente, já que algo está claramente errado, sendo este o comportamento já esperado com qualquer tipo de provedor de serviços. Mas, quando se lida com o intervalo de respostas dentro do intervalo de confiança, com este modelo, o sistema pode agora prever e corrigir comportamento futuro antes que o mesmo ocorra.

Um alto desvio padrão pode indicar que o sistema precisa de uma melhoria de desempenho, assim como uma média pequena com um desvio padrão pequeno pode indicar um cenário passível de economia de energia.

Como este é um modelo genérico, sem análise comportamental de sistema específico, e já que o outro único modelo de *QoE-QoS* para *BDDs* encontrado na literatura (COSTA; FURTADO, 2011) não dá equações específicas e é apenas aplicado a uma base de dados, a comparação de modelos não foi possível, sendo o modelo apresentado um primeiro modelo quantitativo de *avaliação de QoE* baseado em *parâmetros de QoS* matematicamente bem definido para *BDDs*.

A análise comportamental para uma melhor *aplicação de QoE* é uma lacuna que tentamos preencher nas seções seguintes, afinal o modelo genérico ainda necessita de cenários com gatilhos para ajuste de desempenho ou economia de energia. E já que os valores apresentados, como $2 * \sigma$ podem ser extrapolados para qualquer $n * \sigma$, o modelo apresentado é genérico o suficiente para abrir discussão sobre estes parâmetros, métodos de recuperação e economia de energia, bem como os gatilhos para estes métodos.

A aplicação deste modelo irá com certeza impactar em um melhor nível de aceitação de usuário para serviços, mas também requer uma avaliação complexa de comportamento de sistemas pelo lado do

provedor de serviços.

Na próxima seção lidamos com a implementação de avaliações de *QoE*, como apresentadas aqui, a serem utilizadas com sistemas para recuperação de *QoS* por compensação, para ajudar em garantias de *QoS* como definidas em um *ANS*, com seus intervalos de confiança de *QoE*. Porém, frisa-se que a implementação de avaliações apresentada a seguir é apenas uma parte das definições aqui realizadas, de maneira que são necessárias muitas outras considerações para criação de um sistema completo de avaliação e recuperação de *BDDs* com a consideração de *ANSs*.

4.2 MODELO ARQUITETURAL DE RECUPERAÇÃO PARA BASES DE DADOS DISTRIBUÍDAS FOCADO EM R.A.S.

O passo natural da sequência deste estudo tem foco na demonstração do uso do modelo de mapeamento teórico de *QoE* através de *QoS* focado em *BDDs* junto a um modelo de replicação arquitetural, o qual lida com soluções distribuídas. O foco desta seção se dá tanto na busca de soluções de cunho horizontal quanto vertical, de maneira a se focar na escolha dos procedimentos de recuperação mais adequados a serem seguidos de acordo com as necessidades de sistemas de *BDDs*.

Como apresentado nos capítulos anteriores, sobre o modelo teórico, a ausência de definições para *BDDs*, aliada aos conceitos relacionados à redes, e sua natureza holística, levaram a formulação deste novo modelo, de maneira a preencher a lacuna conceitual junto à literatura de *BDDs*, como apresentado no estudo biliométrico realizado no capítulo 3.

Cabe ressaltar que o modelo de avaliação anteriormente apresentado é de abrangência muito maior do que utilizamos neste estudo subsequente, de maneira que, além deste sistema de recuperação focado em *RAS*, muitas outras considerações de avaliação de *QoS*, e mesmo utilização de avaliações específicas de *QoE*, podem ser aplicadas em implementação de sistemas dando garantias à *BDDs*. Este passo apresentado é apenas uma ilustração de algumas das possibilidades que o modelo de avaliação permite.

Este modelo teórico é apresentado na seção 4.1 como uma primeira fundamentação de definições de *QoE* no campo de *BDDs* com métricas tangíveis. Como mencionado anteriormente, assume-se inicialmente que “*QoE* é um conceito que não é limitado apenas ao uso de sistemas ou serviços” (CALLET; MÖLLER; PERKIS, 2013), mas considera-

se a posteriori que o conceito pode ser parcialmente definido como tal para prover a um sistema, ou serviço, a habilidade de autoavaliar sua completude, como apresentado na definição 1.

Com base nestes conceitos e formulações, apresentados na seção 4.1, este estudo segue direções com foco no alcance de soluções para definir um novo modelo arquitetural, focado em confiabilidade e disponibilidade, com efeitos também sobre manutenção, utilizando este primeiro modelo teórico, com arquitetura como apresentada nas subseções seguintes. Considerações sobre as equações de *recursos e qualidade* são também apresentadas para consideração do *framework de fórmulas* como um todo.

Os principais procedimentos comportamentais da arquitetura proposta são apresentados em forma algorítmica na subseção 4.2.2, na forma do algoritmo 1, para ilustrar o caminho a ser trilhado em implementações deste modelo, com um modelo generalizado apresentado na subseção seguinte.

O comportamento da implementação arquitetural para o modelo é também apresentado, utilizando-se uma estrutura multianel, com algumas considerações específicas, para ilustrar as possibilidades de se lidar com uma arquitetura de provimento de serviços com *dois níveis*.

Para fins de simplificação, generalizou-se as equações apresentadas na seção 4.1 da seguinte maneira: as equações 4.2, 4.8, 4.14 e 4.20 são generalizadas na forma da equação 4.26; as equações 4.3, 4.9, 4.15 e 4.21 são generalizadas na forma da equação 4.27; as equações 4.4, 4.10, 4.16 e 4.22 são generalizadas na forma da equação 4.28; as equações 4.5, 4.11, 4.17 e 4.23 são generalizadas na forma da equação 4.29; as equações 4.6, 4.12, 4.18 e 4.24 são generalizadas na forma da equação 4.30; e as equações 4.7, 4.13, 4.19 e 4.25 são generalizadas na forma da equação 4.31.

$$\begin{aligned} QoE_{RAS}^{sistema}(ins|atu|del) &= \frac{TaxaDeUptime}{ANS(GarantiaRASIns|Atu|Del)} & [0; 1] \\ QoE_{RAS}^{sistema}(sel) &= \frac{TaxaDeUptime * TaxaDeAtualização}{ANS(GarantiaRASsel)} & [0; 1] \end{aligned} \quad (4.26)$$

$$\begin{aligned} QoE_{RAS}^{usuário}(ins|atu|del) &= \frac{TaxaDeSucesso}{ANS(GarantiaRASIns|Atu|Del)} & [0; 1] \\ QoE_{RAS}^{usuário}(sel) &= \frac{TaxaDeSucesso * TaxaDeAtualização}{ANS(GarantiaRASsel)} & [0; 1] \end{aligned} \quad (4.27)$$

$$\begin{aligned} QoE_{R\&Q}^{sistema}(ins|atu|del|sel) &= & [0; 1] \\ &= \frac{ANS(GarantiaR\&QIns|Atu|Del|Sel)}{(\text{TempoMedioInicializaçãoInstância} + n_1 * \text{DesvPadInicializaçãoInstância}) + (\text{TempoMedioDeResposta} + n_2 * \text{TempoDeDesvioPadraoDeResposta})} \end{aligned} \quad (4.28)$$

$$QoE_{R\&Q}^{usuário}(ins|atu|del|sel) = \frac{ANS(GarantiaR\&QIns|Atu|Del|Sel)}{TempoDeInicializaçãoDeInstância + TempoDeResposta} \quad [0; 1] \quad (4.29)$$

$$QoE_{total}^{sistema}(ins|atu|del|sel) = \begin{matrix} QoE_{RAS}^{sistema}(ins|atu|del|sel) \\ * QoE_{R\&Q}^{sistema}(ins|atu|del|sel) \end{matrix} \quad [0; 1] \quad (4.30)$$

$$QoE_{total}^{usuário}(ins|atu|del|sel) = \begin{matrix} QoE_{RAS}^{usuário}(ins|atu|del|sel) \\ * QoE_{R\&Q}^{usuário}(ins|atu|del|sel) \end{matrix} \quad [0; 1] \quad (4.31)$$

4.2.1 Análise das métricas

O foco do modelo arquitetural aqui apresentado se dá nas equações 4.26 e 4.27, e em como se definir um modelo arquitetural considerando os parâmetros *RAS*, sendo também apresentada a análise das equações 4.28 e 4.29 para demonstração do impacto do modelo como um todo na predição do comportamento de sistemas e como elas podem ser utilizadas de maneira a evitar comportamento futuro inesperado com análise de dados *sob demanda* - relembrando que o modelo de avaliação de *QoE* apresentado na seção anterior é muito mais abrangente do que a solução para esta consideração em específico como apresentada.

Para a equação 4.26, $QoE_{RAS}^{sistema}(ins|atu|del|sel)$, considerar-se $TaxaDeUptime = taxa$ para o caso de inserção, atualização e deleção, e $TaxaDeUptime * TaxaDeAtualização = taxa$ para o caso de seleção. Este parâmetro *taxa* deve ser um valor individual para cada tipo de operação, sendo aqui considerado como um valor individual por questão de simplificação.

A garantia *RAS* de $ANS(Ins|Atu|Del|Sel)$ é a garantia de completude sobre o valor definido de *taxa*, mas como é uma garantia que considera uma média esperada com um desvio padrão associado, a partir deste ponto vamos considerar este tipo de equação com faixa de garantias separadas em equações como: $ANS_{GarantiaRAS}^{médio}^2$, $ANS_{GarantiaRAS}^{mínimo}$ e $ANS_{GarantiaRAS}^{máximo}$.

Como exemplificação considerara-se uma garantia genérica de:

$$\frac{taxa}{ANS_{GarantiaRAS}^{mínimo}} \geq 1,$$

²Quando não explicitando operações em uma equação, a partir deste ponto, estaremos considerando elas lidando com as quatro operações definidas.

com $ANS_{GarantiaRAS}^{mínimo}$ sendo a média esperada menos uma vez o desvio padrão. Considerando ainda este exemplo de ANS com garantia de 68.2% dos valores estando dentro da faixa da média \pm desvio padrão em uma distribuição normal, temos 84.13% dos valores como estando acima deste mínimo considerado. Note que esta é uma exemplificação genérica, com todas as operações agrupadas, sendo o efeito sobre a garantia de seleção mais forte sobre a $TaxaDeUptime$, já que a mesma é multiplicada pela $TaxaDeAtualização$, e então necessitando de uma $TaxaDeUptime$ mais elevada quando comparado com a mesma equação aplicada à inserção, atualização e deleção.

Assim, temos simplesmente que:

$$taxa \geq ANS_{GarantiaRAS}^{mínimo}.$$

Sendo estas medidas de $QoE_{RAS}^{sistema}^3$ entre 1 e 0.8413 aceitáveis para a unidade de desvio padrão como considerado, mas também medidas que devem disparar ao menos procedimentos de recuperação leves, de maneira a evitar que comportamento inesperado seja alcançado no futuro. Enquanto os resultados da *equação de taxa* estiverem acima do dado $ANS_{GarantiaRAS}^{mínimo}$, as garantias estão sendo mantidas.

Considerando o parâmetro $TaxaDeUptime$, uma primeira solução para se evitar que o valor da *equação de taxa* se aproxime do $ANS_{GarantiaRAS}^{mínimo}$ dado, ou menos, é a de se lidar com escalabilidade horizontal e então se inicializar novas réplicas da base atual. Esta não é uma decisão direta, afinal a métrica de disponibilidade não é capaz de identificar se um serviço está funcional, mas somente se o mesmo pode ser alcançado. Este é um ponto de avaliação complexo, já que o serviço provisto pode estar lidando com uma fila de tamanho limitado para as requisições. Uma melhor solução poderia ser então se lidar com escalabilidade vertical, de maneira a se incrementar o tamanho da fila ou incrementar a capacidade de processamento e acelerar o processo de desenfileiramento, já que este tipo de solução iria impactar de maneira mais amena no parâmetro $TaxaDeAtualização$, que afeta diretamente a avaliação da seleção $QoE_{RAS}^{sistema}$. Mas, deixando-se esta consideração específica quanto ao parâmetro $TaxaDeAtualização$ de lado, a escalabilidade horizontal é a escolha natural para se resolver este tipo de problema a longo prazo.

Outro problema que precisamos levar em consideração, quando escolhendo entre estratégias horizontais e verticais, é o tempo necessá-

³Este é um tipo de garantia complexo, não sendo um valor estático e sim uma garantia combinada de média e desvio padrão.

rio para se replicar bases de dados quando se considerando estratégias horizontais ou a disponibilidade de recursos quando se considerando a estratégia vertical. Com estes problemas considerados, a solução que parece melhor se adequar é a possibilidade de lidarmos com um número limitado de réplicas não prioritárias em um *pool* especial de réplicas dormentes, que podem ser *acordadas* para um nível prioritário, migrando para o *pool* de réplicas disponibilizando serviços.

O gatilho considerado para inicializar processos de manutenção leves, sendo horizontais ou verticais, pode ser o de o parâmetro *taxa* atingir o valor decrescente de $ANS_{GarantiaRAS}^{médio}$, já que o sistema espera trabalhar com este valor médio. Já a chamada de procedimentos de manutenção prioritários é necessária quando o sistema alcança o $ANS_{GarantiaRAS}^{mínimo}$.

Da mesma maneira, *procedimentos de economia de energia* podem ser disparados quando o sistema alcançar $ANS_{GarantiaRAS}^{máximo}$, enviando réplicas para o *pool* não prioritário, ou decrementando os recursos verticalmente.

Já que a replicação horizontal é uma solução com foco mais a longo prazo, esta deve ser utilizada prioritariamente quando se verificado que um sistema não está bem estruturado. As soluções verticais podem ser utilizadas para respostas mais rápidas e para a resolução de problemas pontuais, como os verificados pelo $QoE_{RAS}^{usuário}$, que lida com *TaxaDeSucesso*. Assim, as soluções verticais podem ser utilizadas de maneira a verificar problemas quanto à *uptime* relacionadas a *sucesso* inicialmente, como uma flutuação momentânea, e se a flutuação persistir, o sistema deve disparar procedimentos de replicação horizontal.

Por exemplo, assumindo-se uma *BDD* com s servidores, cada um com uma capacidade abstrata denotada por c , no momento em que $QoE_{RAS}^{sistema}$ alcança o valor de $ANS_{GarantiaRAS}^{mínimo}$, os s servidores devem ser verticalmente escalados para uma capacidade $c = c + n$, onde temos n como o valor de passo incremental de capacidade abstrato.

Existindo um valor médio de capacidade esperado, o nível de capacidade pode ser analisado de tempos em tempos para se decidir se o número de servidores deve crescer. Se sim, o valor da capacidade nos servidores ativos deve então retornar ao valor esperado de c quando uma nova réplica servidora for acordada do *pool* de réplicas dormentes.

Quando a causa da medição inesperada for o *TaxaDeAtualização*, o *access update ratio* pode também ser incrementado para se corrigir o $QoE_{RAS}^{sistema}(sel)$ esperado, como uma solução alternativa.

Para a equação 4.27, o $QoE_{RAS}^{usuário}$, temos uma avaliação que se relaciona à soluções de abordagem vertical, como mencionado anterior-

mente, com ênfase na resolução de problemas relacionados à *TaxaDeSucesso* para problemas identificados como momentâneos.

Considerado-se este tipo de abordagem, se os recursos se estabilizarem em um valor inesperado, digamos acima da capacidade esperada, o sistema deve então disparar procedimentos de escalabilidade horizontal e acordar uma réplica do *pool* dormente. Se o oposto ocorrer, e os recursos se estabilizarem abaixo do nível de capacidade esperada, o sistema pode considerar o disparo de procedimentos de escalabilidade vertical com propósitos de economia de energia, como redução da capacidade de processamento dos nós ativos. Ou, em uma abordagem baseada em capacidade abstrata, reduzir a mesma.

As avaliações pontuais podem também indicar outros problemas relacionados à disponibilidade, que podem ser resolvidos apenas pela inicialização instantânea de procedimentos de natureza horizontal, adicionando-se novos provedores ao *pool* de serviços.

É interessante notar que ambas as estratégias, tanto horizontal como vertical, podem ser inicializadas para resolução de problemas quanto a um serviço específico, porém os procedimentos de escalabilidade irão, em geral, afetar todos os tipos de serviços provistos por um *sistema de BDDs*.

Para as equações 4.28 e 4.29, referidas como $QoE_{R\&Q}^{sistema}$ e $QoE_{R\&Q}^{usuário}$, temos *avaliações temporais*, como mencionado na seção anterior. Neste caso, a solução natural parece se relacionar com procedimentos verticais focados em capacidade de processamento.

A avaliação $R\&Q$ com foco em sistemas lida com valores considerando os desvios padrão, como acordados, e a avaliação com foco no usuário lida com valores momentâneos, transação a transação. A primeira foca na detecção de problemas comportamentais que podem levar o sistema a estados inesperados futuramente, enquanto a segunda foca na detecção de problemas pontuais, *sob demanda*, que necessitam de soluções momentâneas.

Quando se considera a $QoE_{R\&Q}^{sistema}$, temos a garantia de *ANS* de tempo contra o tempo de inicialização de uma instância somado ao desvio padrão acordado para tal, que é basicamente o tempo de se alcançar o servidor, que inclui o tempo gasto em uma fila, com o tempo de resposta da requisição mais o desvio padrão acordado, que é o tempo de processamento da requisição. Com a equação dada é possível se inferir a necessidade de (i) mais processamento, ou (ii) mais provedores de serviço no *pool*, quando o tempo de inicialização não está respondendo como requerido, (iii) mais poder de processamento quando o tempo de resposta não está de acordo com a média esperada, mas

estável, ou (iv) mais provedores de serviços junto ao *pool* de provedores para resolução de estados de alto desvio padrão.

O $QoE_{R\&Q}^{usuário}$ avalia o tempo momentâneo esperado contra o tempo de inicialização de instância mais o tempo de resposta para uma transação específica. Este é normalmente o tempo total de recebimento de resposta para um dado pedido.

Com a avaliação de $QoE_{R\&Q}^{usuário}$, problemas pontuais relacionados a tempos de resposta podem ser identificados. Por exemplo, problemas com um provedor de serviços específico podem ser identificados e o provedor problemático pode ser reiniciado ou substituído por uma das réplicas previstas pelo *pool* de réplicas dormentes.

Se o tempo de espera atingir um limite, o cliente pode sinalizar um possível problema com o servidor de serviços responsável por sua requisição. Se a sinalização persistir, o sistema, que pode ser especificado como um *sistema de gerenciamento de bases de dados (SGBD)*, deve requisitar que o provedor verifique sobre a origem do problema.

4.2.2 Modelo Arquitetural RAS

O modelo apresentado considera duas categorias de *pool* com provedores de serviços: O *pool provedor de serviços*, com os serviços disponíveis e prontos para utilização, sendo provistos pelos servidores principais; e o *pool de backup*, um *pool* com réplicas dormentes, que são acordadas de tempos em tempos, quando as réplicas no *pool provedor de serviços* alcançam *checkpoints*, com mensagens vindas dos *nós de borda*, que são provedores de serviços responsáveis pelo envio de dados de *checkpoints* aos servidores junto ao *pool de backup*.

Esta é uma solução desenvolvida de maneira a evitar operações de grande consumo de tempo para criação de novas réplicas de *BDs* do zero quando lidando com procedimentos de escalabilidade horizontal. Com as réplicas do *pool de backup* já em um estado próximo ao estado das réplicas *online*, é mais rápida a obtenção de novas réplicas no mesmo estado que as réplicas já funcionais junto ao *pool provedor de serviços* quando necessário.

A figura 8 ilustra um exemplo simples com uma implementação de troca de mensagens utilizando o modelo multianel, inspirado no exemplo descrito em (MARANDI; PRIMI; PEDONE, 2012), lidando com dois anéis completos no *pool provedor de serviços*. Os anéis possuem dois nós em área comum para uma exemplificação de tolerância a faltas simples, de maneira que se o nó 2 apresentar problemas, as arestas *a* e

b se tornam ativas, ou se o nó 3 apresentar problemas, as arestas c e d se tornam ativas.

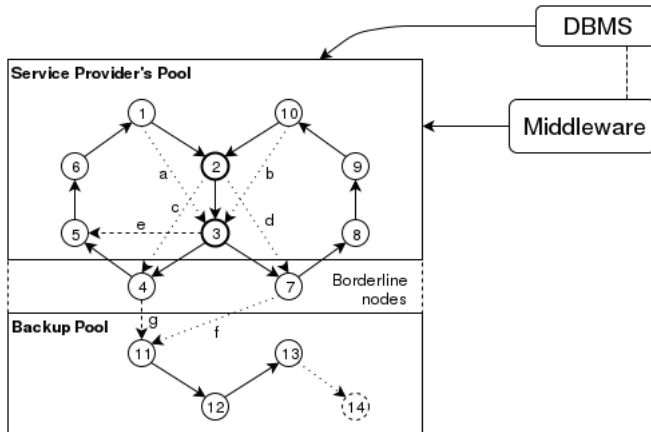


Figura 8: Modelo arquitetural RAS, como publicado em (SOUZA et al., 2016), com implementação de troca de mensagens utilizando o modelo multianel com exemplo de tolerância a uma falta.

Os nós 4 e 7 são chamados de *nós de borda*, porque eles estão no *pool provedor de serviços* e parcialmente no *pool de backup*. O “parcialmente” se dá devido ao fato de a aresta g , ou a aresta f se o nó 4 apresentar problemas, acordar os nós no *pool de backup* ao chamar o procedimento de *checkpoint*. O *pool de backup* é uma abordagem para, até um certo nível, lidar com o problema de consistência, disponibilidade e tolerância a partição (CAP) (GILBERT; LYNCH, 2002) sem afetar o *pool provedor de serviços*, afinal a disponibilidade neste *pool* não é afetada, com a consistência tomando base somente no *pool de backup*.

Quando um novo nó é requisitado junto ao *pool provedor de serviços*, digamos o nó 11, um novo nó replicante deve ser inicializado no *pool de backup*, como o nó 14, e então deve ser adicionado ao semianel de réplicas de *backup*. Este novo nó deve ser replicado com base nos nós do *pool de backup*, para evitar impacto sobre o sistema no lado do *pool provedor de serviços*.

Levando-se em consideração que a arquitetura como um todo é mantida em um ambiente baseado em nuvem, a estrutura pode ser escalada facilmente de maneira vertical, devido à natureza elástica da plataforma hospedeira.

A troca de mensagens multianel é uma abordagem interessante para se evitar excesso de comunicação, mas é apresentada aqui apenas como uma possibilidade de implementação para a arquitetura como definida.

Neste modelo, como apresentado na figura 8, um *middleware* anexado ao *SGBD* analisa o comportamento do sistema como um todo, e controla as decisões relacionadas à realização de procedimentos de manutenção horizontais e verticais.

Note que escalar horizontalmente irá resultar em custos complexos para se manter a consistência de dados, devido ao incremento de latência quando os procedimentos de escalabilidade horizontais tentam manter a consistência (ABADI, 2012). Para se evitar estes custos de latência, a criação de novas réplicas é feita somente com base nas réplicas do lado do *pool de backup*.

Desconsiderando a ilustração relativa à tolerância a faltas e a exemplificação com arquitetura multianel, que não são o foco específico desta definição arquitetural, com base na análise realizada na subseção 4.2.1 e na arquitetura aqui apresentada, os principais procedimentos para lidar com o comportamento esperado no modelo desenvolvido são apresentados junto ao algoritmo 1.

Os procedimentos necessários para se construir este modelo consistem de procedimentos de verificação de garantias mínimas e máximas, verticais e horizontais, de *RAS* e também de procedimentos de sinalização de *checkpoints* alcançados, enviados pelos *nós de borda* para lidar com as réplicas dormentes junto ao *pool de backup*.

4.2.2.1 Procedimentos Algorítmicos

O algoritmo 1 apresenta procedimentos de acordo com as *equações de QoE-QoS* dadas, de maneira a se construir uma solução lidando com os recursos de *sistemas de arquitetura de BDs* de acordo com a variação da *taxa*.

Três procedimentos são apresentados no algoritmo 1:

- O procedimento *Principal(tempo)*, que constantemente verifica a *taxa*, chamando o procedimento $RAS_{mín}(tempo)$ quando o $ANS_{GarantiaRAS}^{mínimo}$ é alcançado, sendo o $RAS_{mín}(tempo)$ um procedimento que inicia mecanismos para prover mais recursos, ou chamando o procedimento $RAS_{máx}(tempo)$ quando o $ANS_{GarantiaRAS}^{máximo}$ é alcançado, de maneira a liberar a utilização de recursos desnecessários;

Algoritmo 1 Procedimentos RAS

```

1: procedimento Principal(tempo)
2:   enquanto verdade faça
3:     se taxa ≤  $ANS_{GarantiaRAS}^{mínimo}$  então
4:       chame  $RAS_{mín}(tempo)$ 
5:     senão se taxa ≥  $ANS_{GarantiaRAS}^{máximo}$  então
6:       chame  $RAS_{máx}(tempo)$ 
7:     senão se  $c \neq c_{original}$  então
8:        $c \leftarrow c_{original}$ 
9:     fim se
10:    espere(tempo)
11:  fim enquanto
12: fim procedimento
13: procedimento  $RAS_{mín}(tempo)$ 
14:   enquanto taxa <  $ANS_{GarantiaRAS}^{médio}$  faça
15:     se  $c = c_{máximo}$  então
16:       se  $\#réplicas = \#réplicas_{máx}$  então
17:         retorna
18:       fim se
19:       Acorde réplica do pool de backup
20:       Adicione a réplica ao pool provedor de serviços
21:        $c \leftarrow c_{original}$ 
22:       Inicie o procedimento de novas réplicas do pool de de backup
23:     senão
24:        $c \leftarrow c + n$ 
25:     fim se
26:    espere(tempo)
27:  fim enquanto
28: fim procedimento
29: procedimento  $RAS_{máx}(tempo)$ 
30:   enquanto taxa >  $ANS_{GarantiaRAS}^{médio}$  faça
31:     se  $c = c_{original}$  então
32:       se  $\#réplicas = \#réplicas_{mínimo}$  então
33:         retorna
34:       senão
35:         Remova uma réplica do pool provedor de serviços
36:       fim se
37:     senão
38:        $c \leftarrow c - n$ 
39:     fim se
40:    espere(tempo)
41:  fim enquanto
42: fim procedimento

```

- o procedimento $RAS_{mín}(tempo)$, que interage enquanto o valor de $QoE_{RAS}^{sistema}$ é mantido abaixo do valor esperado de $ANS_{GarantiaRAS}^{médio}$, aumentando o valor da capacidade c pelo valor predefinido de n , respeitando o intervalo de medida de tempo $tempo$, mantendo o valor da *capacidade estendida* até que a média alcance o $ANS_{GarantiaRAS}^{médio}$ esperado novamente, ou, se o nível de capacidade atingir seu máximo, acordando uma nova réplica do *pool dormente* e resetando o valor de c . Note que esta solução é uma solução conservadora, aumentando a capacidade de c até que $ANS_{GarantiaRAS}^{médio}$, e não $ANS_{GarantiaRAS}^{mínimo}$, seja alcançado;
- e o procedimento $RAS_{máx}(tempo)$, que vai no caminho oposto do procedimento $RAS_{mín}(tempo)$, lidando com um *loop* que inicializa verificando se o valor mínimo aceitável para a capacidade de recursos foi atingido, e se não decrementa o mesmo pelo valor predefinido de n , ou, se atingido, remove uma réplica do *pool provedor de serviços* como um procedimento de economia de energia.

Este é um modelo algorítmico de aparência muito simples, apenas realizando verificações e, se necessário, chamadas a procedimentos de correção, porém capaz de prover garantias de resposta temporais com base em recursos lidando com escalabilidade tanto horizontal quanto vertical.

Este modelo de recuperação arquitetural serve para demonstrar uma primeira utilização das equações apresentadas na seção anterior, de maneira ilustrativa quanto ao ganho que pode ser obtido, não sendo esta a limitação de utilização das equações e conceitos como previamente apresentados.

Na próxima seção analisam-se recursos de maneira estendida, não mais lidando com avaliação de completude temporal com a análise sobre recursos, e sim sobre completude de recursos diretamente, para facilitar o desenvolvimento de experimentos e evitar mapeamentos *difusos*.

4.3 MODELO ARQUITETURAL ESTENDIDO COM FOCO EM RECURSOS DE SISTEMA

Para facilitar a avaliação do modelo via simulação, nesta seção é proposta uma variante do *modelo arquitetural* apresentado na seção anterior. O foco é aqui direcionado ao fornecimento de recursos explicitamente, com base na conceitualização anterior de QoE focada na avaliação de *BDDs*.

Esta variante surge para evitar o mapeamento *difuso* de completude temporal sobre completude de recursos. Uma vez que é difícil lidar com a definição de tempos sem o conhecimento de sistemas específicos, cada qual com suas características particulares, a consideração anterior dificulta a obtenção de resultados via simulação. Por isso, com as definições explícitas de avaliação e garantias sendo feitas sobre recursos, fica mais fácil uma primeira avaliação do modelo via experimentação.

Aqui são apresentadas quatro variantes do algoritmo de decisão da seção anterior, com explicações sobre o porquê da necessidade de *evoluir* o modelo original de maneira a focar em *qualidade*, deixando os pressupostos de *economia de energia* como funcionalidades extra.

Mesmo mantendo a consideração sobre a importância de procedimentos de *economia de energia*, o foco é mantido em *garantias de QoE*, com base em considerações de *QoS*. Portanto, o foco é mantido em garantias comportamentais de sistemas, razão pela qual a *economia de energia* é agora mantida apenas como funcionalidade extra.

O principal ponto deste modelo estendido é a materialização dos conceitos apresentados anteriormente, já que os conceitos de completude temporal, por mais que não sejam completamente abstratos, são dependentes de recursos disponibilizados, é mais simples lidarmos com um modelo dando garantias de recursos. Afinal, os conceitos de completude temporal precisam ser mapeados para completude de recursos, e então evita-se mapeamentos que possam levar a erros de precisão.

Os recursos são agora apresentados em uma conceitualização mais generalizada, com as avaliações podendo ser mais facilmente estendidas em várias dimensões, como poder de processamento, memória volátil e capacidade de armazenamento. Os resultados de simulação, apresentados no capítulo 5, indicam o sucesso da proposta.

4.3.1 Modelo de Completude de Recursos

A figura 9 apresenta um exemplo, de maneira a ilustrar como o modelo de recuperação arquitetural funciona.

Neste exemplo os nós em azul e amarelo: *n01*, *n02*, *n03*, *n04*, *n05*, *n06*, *n07*, *n08*, *n09* e *n10*, são os nós trabalhando junto ao *pool provedor de serviços*, e, portanto, os nós responsáveis por atender as requisições *online*.

Os nós em amarelo: *n04* e *n07*, chamados *nós de borda*, tem a responsabilidade de acordar e enviar as tarefas de modificação, a cada *intervalo de checkpoint*, para os nós dormentes no *pool de backup*.

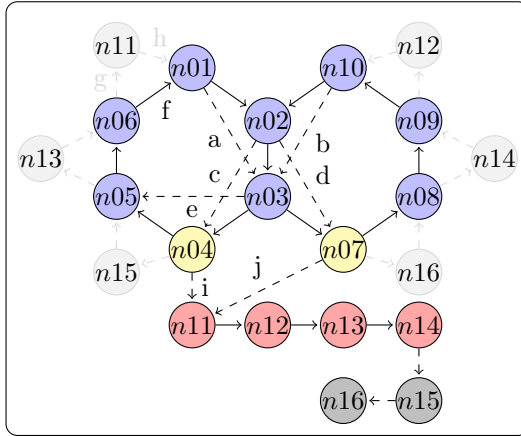


Figura 9: O modelo arquitetural estendido com 10 nós iniciais no *pool* provedor de serviços (em azul e amarelo) e 4 nós iniciais no *pool* de réplicas dormentes (em vermelho), com um máximo de 16 nós em uma implementação de troca de mensagens em multianel com 2 anéis completos, com tolerância a uma falta, e 1 semianel auxiliar de réplicas dormentes.

Em vermelho, os nós: $n11$, $n12$, $n13$ e $n14$, são os nós no *pool de backup*, sendo eles nós dormentes que são acordados a cada *intervalo de checkpoint* por um dos *nós de borda* para receber as modificações do *pool provedor de serviços*. Este *intervalo de checkpoint* pode ser tanto temporal como baseado em número de requisições/modificações junto ao sistema, ou mesmo uma combinação de ambos os fatores.

É estabelecido neste exemplo um número máximo de 16 nós, sendo esta a razão da representação dos nós $n15$ e $n16$ em cinza escuro. Estes são nós ainda não instanciados, mas eles podem ser instanciados pela chamada de um procedimento de replicação completa por um dos nós no *pool de backup*.

Em cinza claro temos as posições dos nós dormentes e também dos nós ainda não instanciados quando os mesmos são inseridos no *pool provedor de serviços*. Quando um nó é adicionado *entre* outros dois nós, a aresta conectando estes nós deixa de existir, sendo adicionadas as arestas conectando estes nós ao novo nó adicionado ao anel. Por exemplo, quando é adicionado o nó $n11$, a aresta f deixa de existir e as arestas g e h são então adicionadas ao anel.

Como apresentado na seção 4.2, o *pool de backup* é uma abor-

dagem para, até um certo grau, lidar com o problema do teorema de consistência, disponibilidade e tolerância a partição (CAP) (GILBERT; LYNCH, 2002), sem impacto sobre o *pool provedor de serviços*.

No exemplo apresentado na figura 9, temos uma estrutura de comunicação multianel, não sendo esta estrutura necessária para utilização deste modelo de recuperação arquitetural, já que os procedimentos são fortemente generalizados. Este exemplo é apresentado para demonstrar como este sistema poderia trabalhar com tolerância a uma falta no modelo multianel, da mesma maneira como apresentado na seção 4.2.

Os anéis no *pool provedor de serviços* têm dois nós em comum para tolerância a uma falta simples, de maneira que se o nó *n02 falhar*, as arestas *a* e *b* se tornam ativas, e se o nó *n03 falhar*, as arestas *c* e *d* se tornam ativas.

A aresta *i*, ou a aresta *j* se o nó *n04 falhar*, é utilizada para acordar o primeiro nó no *pool de backup* sempre que o sistema atingir um *checkpoint*, e atualizar os dados utilizando o anel parcialmente conectado do *pool de backup*, chamando o procedimento de *checkpoint*. Este nó recém despertado acorda o próximo nó, enviando os novos dados e então retorna ao estado de dormência, e assim por diante.

Quando um novo nó é necessário junto ao *pool provedor de serviços*, como o nó *n11* por exemplo, uma nova réplica deve ser inicializada no *pool de backup*, como o nó *n15*, e então adicionada ao semianel de réplicas de *backup*. Este novo nó deve ser replicado com base nos nós do *pool de backup*, de maneira a evitar impacto sobre o sistema junto ao *pool provedor de serviços*.

Como pode ser observado na figura 9, este exemplo adiciona novos nós seguindo o contrafluxo de mensagens alternadamente entre os anéis, evitando os nós de interconexão entre os mesmo. Esta é apenas uma exemplificação, já que a arquitetura aqui apresentada é bastante genérica.

Da mesma maneira, é definido que a distribuição de requisições segue o contrafluxo de maneira alternada entre os anéis, como apresentado na tabela 18. Note que o índice indica em qual rodada de *adição de nós* o nó está presente no *pool*, de maneira que, se a rodada ainda não foi alcançada, a mensagem é enviada ao próximo nó nesta cadeia utilizando uma sequência de *round-robin*.

$n01_0$	\rightarrow	$n10_0$	\rightarrow	$n11_1$	\rightarrow	$n12_2$	\rightarrow
$n06_0$	\rightarrow	$n09_0$	\rightarrow	$n13_3$	\rightarrow	$n14_4$	\rightarrow
$n05_0$	\rightarrow	$n08_0$	\rightarrow	$n15_5$	\rightarrow	$n16_6$	\rightarrow
$n04_0$	\rightarrow	$n07_0$	\rightarrow	$n03_0$	\rightarrow	$n02_0$	

Tabela 18: Fila de distribuição de requisições de operações.

O algoritmo 2 apresenta uma abordagem mais generalizada do algoritmo apresentado na seção 4.2, focando em gerenciamento de recursos. Ele também apresenta quatro modos de operação, cada um dos quais com seu comportamento e utilidade particulares.

Algoritmo 2 Procedimentos de recuperação de algoritmo para gerenciamento de recursos em sistemas distribuídos

```

1: procedimento Principal(tempo, tipo, passo)
2:   enquanto verdade faça
       $\frac{\sum_{i=1}^{\#rélicas} r_{élicas}[i].c_{livre}}{\#rélicas} < passo$  então
3:     se  $\frac{\sum_{i=1}^{\#rélicas} r_{élicas}[i].c_{livre}}{\#rélicas} < passo$  então
4:       chama  $RAS_{mín}(tempo, tipo, passo)$ 
       $\frac{\sum_{i=1}^{\#rélicas} r_{élicas}[i].c_{usado}}{\#rélicas}$  então
5:     senão se  $c - passo > \frac{\sum_{i=1}^{\#rélicas} r_{élicas}[i].c_{usado}}{\#rélicas}$  então
6:       chama  $RAS_{máx}(tempo, tipo, passo)$ 
7:     fim se
8:     espere(tempo)
9:   fim enquanto
10: fim procedimento
11: procedimento  $RAS_{mín}(tempo, tipo, passo)$ 
       $\frac{\sum_{i=1}^{\#rélicas} r_{élicas}[i].c_{livre}}{\#rélicas} < passo$  faça
12:   enquanto  $\frac{\sum_{i=1}^{\#rélicas} r_{élicas}[i].c_{livre}}{\#rélicas} < passo$  faça
13:     se  $c = c_{máx}$  então
14:       se  $\#rélicas = \#rélicas_{máx}$  então
15:         retorne
16:       fim se
17:       se tipo = otimista então
18:          $c \leftarrow C_{original}$ 
19:       senão se tipo = balanceado então
20:          $c \leftarrow \lceil \frac{\#rélicas * c}{\#rélicas + 1} \rceil$  (arredondado em passo)
21:       fim se
22:       Acorde uma réplica do pool de backup
23:       Adicione réplica ao pool provedor de serviços
24:       Inicie o procedimento de nova réplica no pool de backup

```

```

25:      senão
26:           $c \leftarrow c + \text{passo}$ 
27:      fim se
28:      espere(tempo)
29:  fim enquanto
30: fim procedimento
31: procedimento  $RAS_{m\acute{a}x}(\text{tempo}, \text{tipo}, \text{passo})$ 
      
$$\frac{\sum_{i=1}^{\#r\acute{e}plicas} r\acute{e}plicas[i].c_{usado}}{\#r\acute{e}plicas}$$

32:      enquanto  $c - \text{passo} > \frac{\sum_{i=1}^{\#r\acute{e}plicas} r\acute{e}plicas[i].c_{usado}}{\#r\acute{e}plicas}$  faça
33:          se  $c = c_{original}$  então
34:              se  $\#r\acute{e}plicas = \#r\acute{e}plicas_{m\acute{in}}$  então
35:                  retorne
36:              senão
37:                  se tipo = balanceado então
38:                       $c \leftarrow \lfloor \frac{\#r\acute{e}plicas * c}{\#r\acute{e}plicas - 1} \rfloor$  (arredondado em passo)
39:                      senão se tipo = não-otimista então
40:                           $c \leftarrow c_{m\acute{a}x}$ 
41:                      senão se tipo = não-otimista aprimorado então
42:                           $c \leftarrow \lceil \frac{\#r\acute{e}plicas * c}{\#r\acute{e}plicas - 1} \rceil$  (arredondado em passo)
43:                      fim se
44:                      Remove réplica do pool provedor de serviços
45:                  fim se
46:              senão
47:                   $c \leftarrow c - \text{passo}$ 
48:              fim se
49:              espere(tempo)
50:          fim enquanto
51: fim procedimento

```

A abordagem de gerenciamento de recursos surgiu como uma alternativa à completude de tempo de *QoE-QoS* apresentada na seção anterior, com base no fato de que os tempos de resposta são estritamente conectados com a disponibilidade de recursos. Desta maneira, evita-se o mapeamento de completude temporal para completude de recursos, necessária quando se lida com a abordagem anterior, tornando o sistema mais robusto em relação às garantias dadas.

Note que, por questão de simplificação, lidamos com a avaliação relativa ao *passo* de acréscimo de recursos, sempre garantindo que o sistema esteja um *passo* acima do requerido por nó, não mais lidando com as garantias de *ANS* de completude temporais. Desta maneira garantimos que o sistema tenha sempre, pelo menos, a quantidade de

recursos necessárias por nó mais uma vez este valor de *passo*.

4.3.1.1 Modos de Operação

Considerando-se o exemplo como apresentado na figura 9, e, portanto, um exemplo utilizando a estrutura multianel com tolerância a falta de um nó, com os novos nós seguindo o contrafluxo de mensagens alternadamente entre os anéis e evitando os nós de interconexão entre anéis, e a distribuição de requisições seguindo o contrafluxo de mensagens alternadamente entre anéis, como apresentado na tabela 18, a partir deste ponto se abstrai o posicionamento de nós quando os adicionando, e apenas é feita referência à *adicionar um nó ao pool*.

Este algoritmo trabalha com quatro modos de operação, sendo eles: (i) modo *otimista*, que é basicamente o algoritmo como apresentado inicialmente na seção 4.2, com a muito arriscada consideração de se reduzir a capacidade recursos disponibilizados em todos os nós ao mínimo no momento em que se adiciona um novo nó instanciado - razão exata pela qual é chamado de *otimista* -, de maneira que, quando incrementado horizontalmente, o *pool* decrementa verticalmente, em uma consideração individual de nós, e faz o incremento vertical como necessário posteriormente. Este modo decrementa horizontalmente se o total de recursos for mantido acima, ou igual, a quantidade antes do decremento; (ii) modo *não-otimista*, que é um modo trabalhando na direção oposta do modo *otimista*, sempre mantendo todos os nós com o máximo de recursos possível quando adicionando um novo nó, e então decrementando o *pool* verticalmente segundo necessidade. Este modo também modifica os recursos individuais dos nós à capacidade máxima quando decrementando horizontalmente, de maneira a estar preparado para lidar com flutuações de requisição de maneira *não otimista*; (iii) modo *balanceado*, um modo que tenta estar entre os modos *otimista* e *não-otimista*, balanceando os recursos verticalmente quando incrementando ou decrementando o *pool* horizontalmente, de maneira a tentar manter o total de recursos disponíveis em pelo menos o mesmo número de recursos disponíveis antes do incremento ou decremento horizontal; e (iv) modo *não-otimista aprimorado*, que é uma variação do modo *não-otimista* focada em manter o total de recursos disponibilizados em um mínimo possível acima do valor disponibilizado antes de um decremento horizontal, como uma consideração extra de *economia de energia*.

4.3.1.2 Procedimentos Algorítmicos

É possível observar que o algoritmo 2 provê procedimentos de acordo com as considerações de recursos *QoE-QoS* dadas por *passo*, de maneira a construir uma solução gerenciando os recursos de um *sistema distribuído* de acordo com a *capacidade de recursos livre dos nós*

$(\frac{\sum_{i=1}^{\#réplicas} réplicas[i].c_{livre}}{\#réplicas})$ como *mantida dentro* da garantia dada.

Os três procedimentos apresentados no algoritmo 2: (i) procedimento *Principal(tempo, tipo, passo)*, que constantemente verifica se os recursos disponibilizados estão um mínimo acima do necessário, chamando o procedimento $RAS_{min}(tempo, tipo, passo)$ quando este mínimo é alcançado, sendo o procedimento $RAS_{min}(tempo, tipo, passo)$ um procedimento que inicia mecanismos que providenciam mais recursos. Ou chamando o procedimento $RAS_{max}(tempo, tipo, passo)$ quando reduzir a capacidade não causa risco do sistema alcançar um total de recursos inferior ao requerido pela garantia de recursos dada. Note que trabalhando com qualquer um dos quatro modos a redução de recursos é sempre mais cuidadosa, tentando sempre manter ao menos uma vez o valor de *passo* acima dos recursos como necessários em uma consideração de nós individuais; (ii) o procedimento $RAS_{min}(tempo, tipo, passo)$, que interage enquanto o valor médio dos recursos livres de nó são mantidos abaixo do valor esperado, incrementando o valor de capacidade *c* pelo valor definido de *passo*, respeitando o intervalo de medição de *tempo*, mantendo o valor de *capacidade estendida* até que o valor esperado seja alcançado novamente, ou, se o nível de capacidade atingir seu máximo valor possível, acordando uma nova réplica do *pool dormente* com as considerações específicas do modo; e (iii) o procedimento $RAS_{max}(tempo, tipo, passo)$, que executa na direção oposta do procedimento $RAS_{min}(tempo, tipo, passo)$, lidando com um laço que primeiro verifica se o valor mínimo aceitável para capacidade de recursos foi alcançado, e, se não, decrementa o valor pelo definido valor de *passo*, ou se sim, remove uma réplica do *pool provedor de serviços* como um procedimento de *economia de energia*, com as considerações específicas definidas pelo modo escolhido.

Com este modelo definido com base em recursos, parte-se então para a implementação de um ambiente de experimentação, descrito na seção 4.5, para posterior análise de dados.

4.4 CONSIDERAÇÕES SOBRE A MODELAGEM

Uma vez que a revisão bibliométrica providenciou uma visão sistemática sobre os tópicos abordados, sentiu-se a necessidade de definições exatas de métricas de *QoS* para *BDDs*, resultando nas definições realizadas neste capítulo. Com os indicadores estatísticos demonstrando que a abordagem de *QoE* parece ter sido negligenciada neste campo, e considerando a possibilidade de mapeamento direto de *QoE* em como o usuário percebe um serviço em termos de *parâmetros de QoS* (LAGHARI et al., 2011), este mapeamento é então realizado na primeira seção deste capítulo.

De modo a evitar se lidar com “problemas psicológicos subjetivos e aspectos cognitivos humanos que são tipicamente desconsiderados e determinam diretamente a *QoE*” (LAGHARI et al., 2011), é apresentada a conceitualização de *QoE* como um mapeamento direto para definições de *QoS* segundo um *ANS*. O motivo para tal é o fato de que em sistemas de *BDs* a avaliação pode realmente ser ligada mais a *sistemas* do que a *usuários*, em especial ao se utilizar definições em um *ANS*.

Torna-se possível, com este primeiro modelo matemático bem definido, trilhar o caminho em direção a *sistemas autorrecuperáveis de QoE-QoS*, partindo-se, então, para a formulação de tais sistemas.

Na segunda seção deste capítulo apresenta-se um primeiro modelo arquitetural, indicando a possibilidade de utilização de formulações complexas, lidando-se com avaliações como apresentadas na primeira seção. Desta maneira é possível prever, com o intuito de evitar, comportamento inesperado do sistema antes que o mesmo atinja estados não desejados, com um modelo arquitetural de confiabilidade, disponibilidade e manutenibilidade sendo apresentado.

As soluções propostas são baseadas em escalabilidade horizontal e vertical aplicadas a *sistemas de BDDs* e podem ser diretamente implementadas sobre um *SGBD* para evitar que tais estados sejam alcançados. Não sendo estes conceitos novidade *per se*, mas bastando para preencher este vazio conceitual sobre *arquitecturas* na área de *bases de dados*.

Um primeiro algoritmo é apresentado nesta segunda seção, com alguns procedimentos considerando um comportamento genérico esperado, de maneira a ilustrar a sequência do estudo em direção à terceira seção do capítulo.

Na terceira seção, quatro modos de operação são apresentados para se lidar com um modelo de recuperação arquitetural focado em *recursos*, cada um dos mesmos com suas especificações e uso particula-

res. A decisão sobre qual modo utilizar deve envolver conhecimento do sistema sobre o qual se deseja aplicar tal modelo, sendo recomendado testes de performance sobre o sistema para verificar situações específicas e como cada um dos modelos irá se comportar.

Com o modelo e arquiteturas apresentadas, parte-se, na próxima seção, para a definição de um simulador, de maneira a permitir considerações de sistemas distribuídos sem a necessidade de construção de um ambiente de tal porte.

4.5 AMBIENTE DE EXPERIMENTAÇÃO

Para facilitar os experimentos e permitir que as simulações fossem realizadas independente de sistemas de *BDs* específicos, um ambiente de simulação generalizado foi desenvolvido.

De maneira a permitir a realização de simulações com considerações relativas a sistemas distribuídos, as operações possuem então definição de tempo discretas, independente de unidade.

Este ambiente foi desenvolvido com o foco específico de implementação do algoritmo 2, apresentado na seção 4.3, com pequenas variações, relacionadas a cálculo de recursos, surgidas do modelo como implementado e sua respectiva análise experimental.

A variação que mais chama a atenção é o arredondamento “para baixo” quando reduzindo recursos verticalmente ao se adicionar nós horizontalmente no *modo balanceado*. Através de experimentação se notou que esta variação fornece o comportamento como esperado para tal modo. O arredondamento “para cima”, utilizado no *modo não-otimista aprimorado*, fornece um comportamento realmente mais próximo do *modo não-otimista*, com a redução do pico de consumo de recursos, enquanto que o arredondamento “para baixo” dá ao *modo balanceado* um comportamento mais próximo do *modo otimista*.

4.5.1 Algoritmo de Simulação

O algoritmo de simulação utilizado no ambiente de experimentação desenvolvido é apresentado em sua forma generalizada como o algoritmo 3. Este é um algoritmo composto por três procedimentos simples: (i) *PróximoTempo(recursos, passo, mínSistema, máxSistema, nósAtivos, nósDormentes)*, responsável pela verificação de necessidade de aumento ou redução de recursos junto ao sistema; (ii) *Acre-*

centa(recursos, *mínSistema*, *máxSistema*, *passo*, *nósAtivos*, *nósDormentes*), que incrementa recursos horizontal ou verticalmente segundo definição do modo escolhido; e (iii) *Reduz*(recursos, *mínSistema*, *máxSistema*, *passo*, *nósAtivos*, *nósDormentes*), que decrementa recursos horizontal ou verticalmente segundo definição do modo escolhido.

Os parâmetros de entrada do algoritmo são: (i) *recursos*, referente à quantidade de recursos requerida no referido tempo; (ii) *passo*, que é o valor de passo de incremento e decremento de recursos utilizado pelo sistema em nós individuais; (iii) *mínSistema*, que indica o valor mínimo de recursos que um nó individual pode ter disponibilizado no sistema; (iv) *máxSistema*, que indica o valor máximo de recursos que um nó individual pode ter disponibilizado no sistema; (v) *nósAtivos*, um array com todos os nós ativos do sistema; e (vi) *nósDormentes*, um array com todos os nós dormentes do sistema.

Algoritmo 3 Algoritmo de Simulação

```

1: procedimento PróximoTempo(recursos, passo, mínSistema, máxSistema,
   nósAtivos, nósDormentes)
2:   máxRecurso  $\leftarrow$  nósAtivos[0].recursosMáximo
3:   limitação  $\leftarrow$  #nósAtivos * (máxRecurso - passo) + passo
4:   se recursos  $\geq$  limitação então
5:     chama Acrecenta(passo)
6:   senão se  $\forall$  nósAtivos.recursosUsados + passo < máxRecurso então
7:     chama Reduz(passo)
8:   fim se
9:   Distribui recursos pelos nósAtivos utilizando fila round-robin
10: fim procedimento
11: procedimento Acrecenta(recursos, mínSistema, máxSistema, passo,
   nósAtivos, nósDormentes)
12:   máxRecurso  $\leftarrow$  nósAtivos[0].recursosMáximo
13:   se máxRecurso < máxSistema então
14:      $\forall$  nósAtivos.recursoMáximo  $\leftarrow$  nósAtivos.recursoMáximo + passo
15:   senão
16:     nósAtivos.adiciona(nósDormentes.remove(0))
17:   se modo = otimista então
18:      $\forall$  nósAtivos.recursoMáximo  $\leftarrow$  mínSistema
19:   senão se modo = balanceado então
20:      $\forall$  nósAtivos.recursoMáximo  $\leftarrow$   $\lceil \frac{\textit{máxRecurso} * (\textit{\#nósAtivos} - 1)}{\textit{\#nósAtivos}} \rceil$  (ar-
       redondado em passo)
21:   fim se
22:   fim se
23: fim procedimento

```

```

24: procedimento Reduz(recursos, mínSistema, máxSistema, passo,
    nósAtivos, nósDormentes)
25:   máxRecurso  $\leftarrow$  nósAtivos[0].recursosMáximo
26:   se máxRecurso > mínSistema então
27:      $\forall$ nósAtivos.recursoMáximo  $\leftarrow$  nósAtivos.recursoMáximo -
        passo
28:   senão
29:     nósDormentes.adiciona(nósAtivos.remove(0))
30:   se modo = (balanceado) então
31:      $\forall$ nósAtivos.recursoMáximo  $\leftarrow$ 
         $\left\lceil \frac{\textit{máxRecurso} * (\# \textit{nósAtivos} + 1)}{\# \textit{nósAtivos}} \right\rceil$  (arredondado em passo)
32:   senão se modo = não-otimista então
33:      $\forall$ nósAtivos.recursoMáximo  $\leftarrow$  máxSistema
34:   senão se modo = não-otimista aprimorado então
35:      $\forall$ nósAtivos.recursoMáximo  $\leftarrow$ 
         $\left\lceil \frac{\textit{máxRecurso} * (\# \textit{nósAtivos} + 1)}{\# \textit{nósAtivos}} \right\rceil$  (arredondado em passo)
36:   fim se
37: fim se
38: fim procedimento

```

Note que nas linhas 2, 12 e 25 do algoritmo obtemos o valor máximo de recursos momentâneo de apenas um nó porque neste exemplo simplificado consideramos que a limitação é a mesma para todos os nós.

Com esta definição algorítmica, os resultados apresentados no capítulo 5 podem ser reproduzidos com qualquer implementação ou mesmo simulação como se deseje realizar.

4.5.2 Considerações sobre o Ambiente

Este é um ambiente com definições simples, baseado no algoritmo 3, formalizado para permitir não apenas a criação de dados para análise, mas também facilitar a reprodução de obtenção de resultados.

O ambiente é apresentado de forma algorítmica para permitir eventual reprodução, independente de linguagem de programação selecionada. Assim, trabalhos futuros podem ser comparados de maneira mais simplificada com o apresentado nesta tese. Sendo digno de nota que a implementação realizada segue de maneira exata o algoritmo como apresentado anteriormente.

A utilização de tempos discretos, sem definição de unidade, é

também utilizada para demonstrar resultados de maneira completamente independente de considerações relativas a sistemas específicos, podendo o modelo ser abstraído para situações desejadas, sem se forçar nenhuma direção a ser seguida.

Porém note que a implementação realizada segue exatamente o modelo como apresentado na figura 9, ou seja, temos um modelo multianel com dois anéis, com um total de 10 nós iniciais no *pool provedor de serviços*, 4 nós no *pool de réplicas dormentes*, num total de 16 nós máximos.

O ambiente criado segue também os parâmetros como definidos na subseção 4.3.1, com a mesma distribuição de requisições seguindo o contrafluxo de maneira alternada entre anéis, como definido na tabela 18. A adição de nós seguindo este contrafluxo também se mantém.

5 ANÁLISE EXPERIMENTAL

Uma análise experimental foi realizada utilizando o ambiente de experimentação apresentado na seção 4.5, aplicado sobre o modelo apresentado na seção 4.3, o qual foi criado exclusivamente para facilitar a aplicação de experimentação.

5.1 RESULTADOS DE SIMULAÇÃO

Todos os quatro modos de operação têm o comportamento particular de acordar prioritariamente os nós, como necessário, mas sendo mais cuidadosos quando decidindo em colocá-los para dormir. A razão é evitar que cenários de excesso de consumo de energia sejam alcançados, com variações de requisição causando uma situação contínua de *acordar-adormecer* de nós, o que não é de maneira alguma bom para o sistema.

Para as simulações são considerados 201 *tempos* discretos, sem nenhuma especificação particular de métricas, apenas considerando que a cada um destes tempos o sistema é requerido a prover uma quantidade específica de recursos. Para demonstrar o comportamento do sistema, o total de recursos requeridos é incrementado pelo valor de 1000 a cada 5 tempos, dando tempo ao sistema para fazer as modificações como necessárias até que se chegue a um valor estabilizado. A quantidade de recursos requerida é incrementada de 0 a 20000 e então decrementada de 20000 a 0, como demonstrado nas figuras 10, 11, 12 e 13 com a linha em roxo, a qual inicializa e finaliza no valor de 0 recursos requisitados.

O sistema de simulação exemplo é composto por nós com uma capacidade mínima abstrata de 1000, e uma capacidade máxima de 1200, com a capacidade de incremento de 100 a cada passo. Inicialmente, o *pool provedor de serviços* tem a capacidade de 10 nós, com a capacidade máxima de 4 nós no *pool de backup*. O número máximo de nós provisto pelo sistema é de 16, da mesma maneira como apresentado na figura 9.

Na figura 10 pode ser verificado a principal implicação do algoritmo *otimista* original acompanhando-se a linha em azul, a qual representa a capacidade de recursos disponibilizados: quando se adiciona um novo nó ao *pool* são exigidos vários incrementos verticais até que estabilize o valor referente a recursos disponibilizados.

Uma abordagem que é melhor aplicada a casos onde se tem eco-

nomia de recursos como principal fator. Esta é uma abordagem que tenta garantir que a quantidade de recursos consumida seja a menor possível dando as garantias previstas, de maneira a se considerar a possibilidade de redução de necessidade antes de estabilização dos valores pedidos.

O modelo *otimista* é recomendado para dimensões de recursos que sejam de difícil inicialização, ou casos em que a economia de recursos seja crucial.

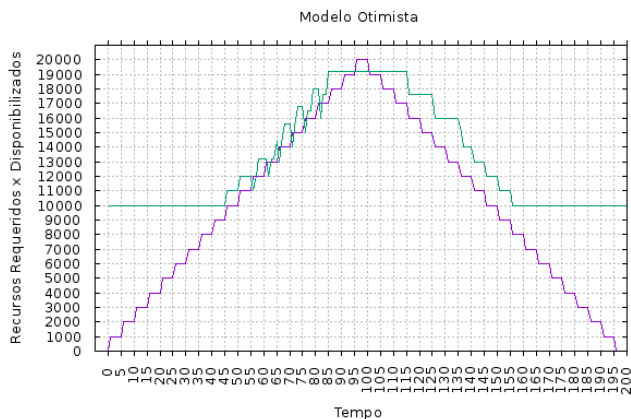


Figura 10: Recursos requeridos e disponíveis do modelo otimista.

Quando se incrementa os recursos necessários no passo especificado de 1000, a diferença entre o modelo *balanceado*, apresentado na figura 11, e os dois modelos *não-otimistas*, apresentados nas figuras 12 e 13, é mínimo, não sendo possível verificar claramente nos gráficos apresentados. A explicação para tal comportamento é que quando se necessita de mais recursos, o sistema simplesmente os adiciona.

Em contraste em relação ao modelo *otimista*, estes outros três modelos tem como foco simplesmente prover rapidamente os recursos como requeridos, sem se ater a economia de recursos em caso de variação durante o processo de acréscimo.

Quando se decrementa os recursos necessários com o passo especificado de 1000, os modelos *otimista* e *balanceado*, nas figuras 10 e 11, respectivamente, tem um comportamento muito similar, já que eles apenas decrementam quando possível, sem considerações extras para o caso *otimista* e com o balanceamento utilizado no modo *balanceado* sendo imperceptível nos gráficos quando comparado ao caso *otimista*.

Estes dois modelos não levam em consideração a possibilidade de variação no sentido contrário quando do momento de decréscimo de recursos, de maneira a serem recomendados para casos de menor variação esperada em relação à decréscimo, como na dimensão de capacidade de armazenamento, por exemplo.

O modelo *balanceado* é o recomendado por padrão pelo fato de manter sempre o valor disponibilizado muito próximo, mas sempre superior, ao valor requerido.

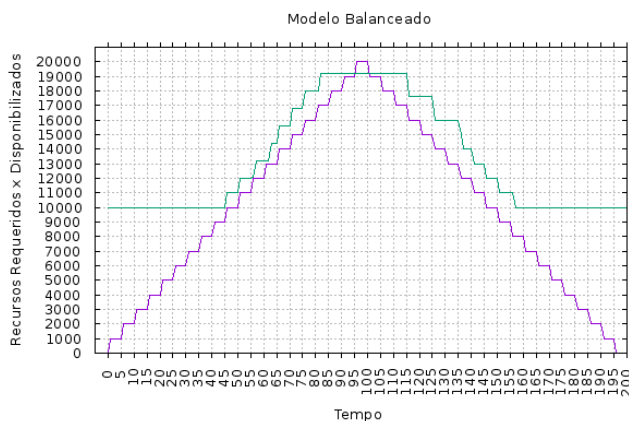


Figura 11: Recursos requeridos e disponíveis do modelo balanceado.

Em casos que ainda não se tem ideia do que se espera de uma dimensão sendo avaliada, deve-se iniciar os procedimentos de avaliação com o modo *balanceado*.

Este modo tendo surgido, na verdade, após a comparação dos modos *otimista* e *não otimista*, como uma definição de tentativa de manutenção de recursos em um valor aceitável para casos em que não ocorre uma necessidade de adição ou redução drástica de recursos em espaços pequenos de tempo.

De certa maneira o modo *balanceado* lida com o incremento de maneira muito próxima do modo *não-otimista*, e o decremento de maneira muito próxima do modo *otimista*.

O decremento do modelo *não-otimista*, como demonstrado na figura 12, tem o comportamento particular de escalar verticalmente os nós ao valor máximo de recursos possíveis quando faz a redução horizontal, causando assim várias reduções verticais em sequência antes a alcançar a estabilidade.

Este sendo um modelo preparado para casos de “mudança de direção” quando de decremento de recursos, ou seja, esperando a possibilidade de necessidade de acréscimo logo após um decréscimo.

O modo *não-otimista* é recomendado para avaliação de dimensão de recursos que possam ser escalados facilmente, sem grandes perdas relativas a variação de capacidade por um curto espaço de tempo, mas que possam ter um impacto sobre a avaliação de serviços se requisitados sem a quantidade de recursos disponibilizados, como a dimensão de poder de processamento.

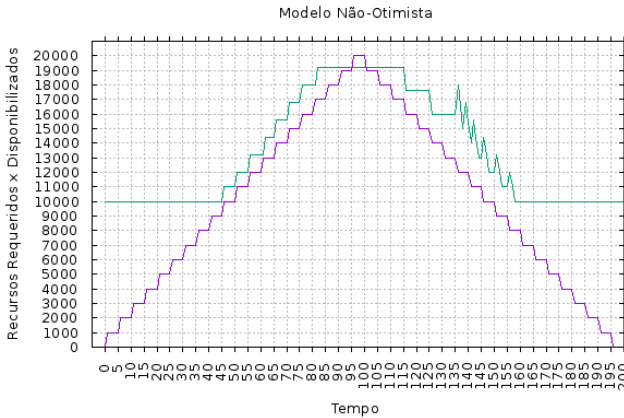


Figura 12: Recursos requeridos e disponíveis do modelo não-otimista.

É claro que este é um modo com considerações exageradas, não percebido facilmente na exemplificação por estarmos trabalhando com nós que tem sua quantidade de recursos variando apenas de 1000 a 1200. Isso fica mais claro se imaginarmos nós variando de 1000 a 10000, quando a variação seria gigantesca em caso de decremento, ou o impacto de uma variação desta magnitude sobre uma dimensão como capacidade de armazenamento, caso em que explicitamente não se recomenda este tipo de abordagem.

Tentando evitar o excesso de variações verticais, o modelo *não-otimista aprimorado*, como demonstrado na figura 13, define a quantidade de recursos dos nós de maneira a garantir apenas um *passo* acima do número de recursos provistos antes de quando reduzindo o *pool* horizontalmente.

Sendo a abordagem do modelo *não-otimista aprimorado* a indicada para se lidar com avaliações de dimensão de capacidade de pro-

cessamento, nas quais não se corre risco de necessidade de aumento exponencial de carga.

Porém frisa-se que o modelo *não-otimista* tem também sua aplicação, como por exemplo ambientes *clusterizados* com alta demanda eventual de capacidade de processamento.

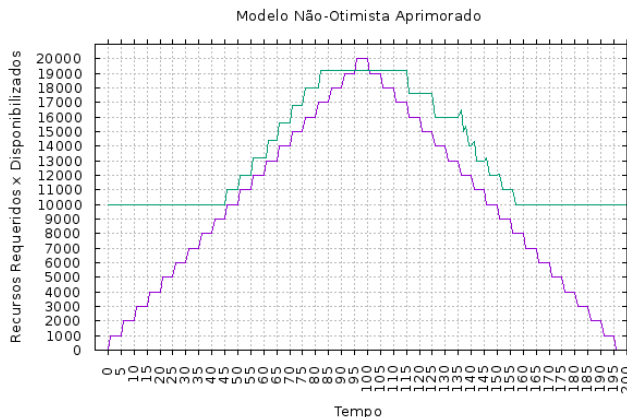


Figura 13: Recursos requeridos e disponíveis do modelo não-otimista aprimorado.

Cada modo tem suas particularidades, e mesmo que, quando avaliados sob certa perspectiva, pareçam ter sua utilidade sobreposta por outro, cada um deles, justamente devido a estas particularidades, possui aplicação específica para avaliação de certas dimensões de recursos em casos especiais.

E mesmo estes ambientes de avaliação parecendo simples, deve-se levar em conta que a utilização de modelos diferenciados para avaliar múltiplas dimensões, quando lidando-se com um único serviço, podem levar a um alto grau de complexidade quando da definição do ambiente de avaliação total de tal serviço.

Os recursos disponíveis nos nós individuais, como demonstrado nas figuras 14, 15, 16 e 17, demonstram que o modelo *otimista*, apresentado na figura 14, tem uma variação vertical maior quando lidando com o incremento de requisições; o modelo *não-otimista*, apresentado na figura 16, tem uma variação vertical maior quando o número de requisições decrementa; o modelo *não-otimista aprimorado*, apresentado na figura 17, tem o mesmo número de variações do modelo *não-otimista*, mas com valores menores; e o modelo *balanceado*, apresentado na figura

15, é o mais estável em termos de variações verticais.

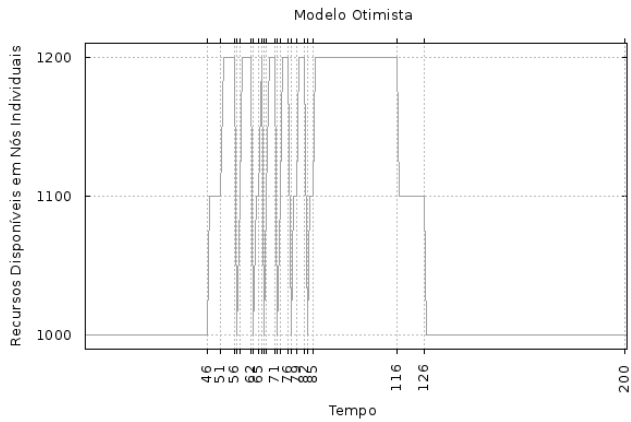


Figura 14: Variação de recursos individuais de nós no modelo otimista para alcançar os valores de recursos requisitados

Cabe salientar que, no comportamento apresentado nas figuras 14, 15, 16 e 17, a dimensão considerada a ser avaliada, seja poder de processamento, memória volátil ou capacidade de armazenamento, pode requerer um modo específico para trabalhar como o esperado.

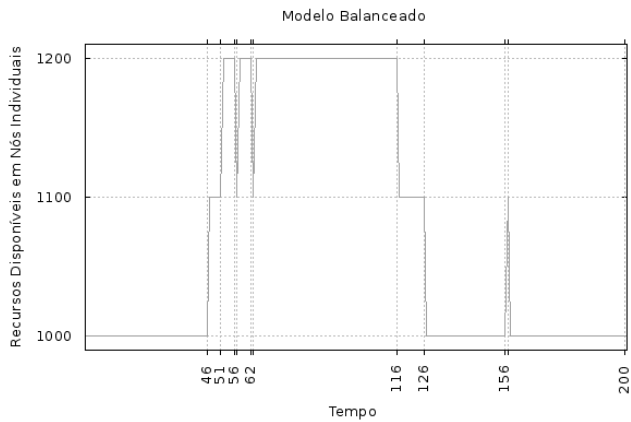


Figura 15: Variação de recursos individuais de nós no modelo balanceado para alcançar os valores de recursos requisitados

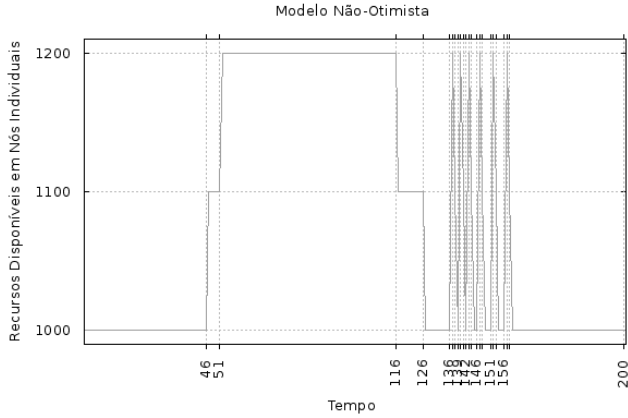


Figura 16: Variação de recursos individuais de nós no modelo não-otimista para alcançar os valores de recursos requisitados

Nas figuras 18, 19, 20 e 21, percebe-se que a escalabilidade horizontal é muito próxima nos quatro modelos apresentados, existindo pequenas variações relacionadas a maneira como o sistema lida com a escalabilidade vertical em cada modo.

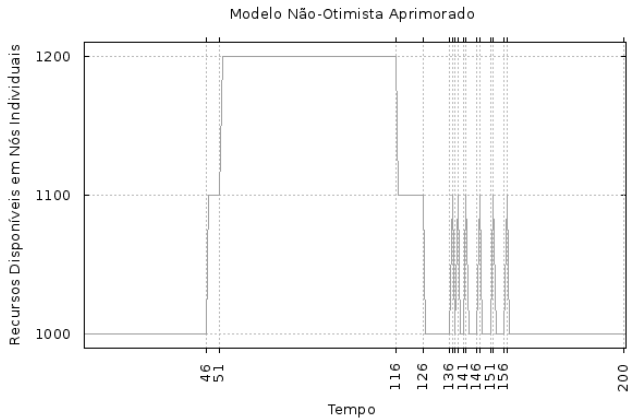


Figura 17: Variação de recursos individuais de nós no modelo não-otimista aprimorado para alcançar os valores de recursos requisitados

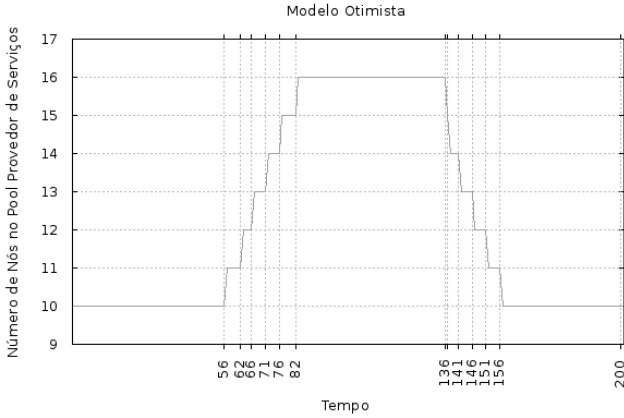


Figura 18: Número de nós no pool provedor de serviços no modelo otimista em resposta aos recursos requisitados

Os experimentos aqui apresentados apenas consideraram o incremento e decremento controlado em um cenário passo a passo, de maneira que a escolha particular de cada modo deve ser analisada com cuidado quando for selecionado um modelo para aplicar a um tipo específico de dimensão a ser avaliada para tomada de decisões de escalabilidade em *sistemas distribuídos*.

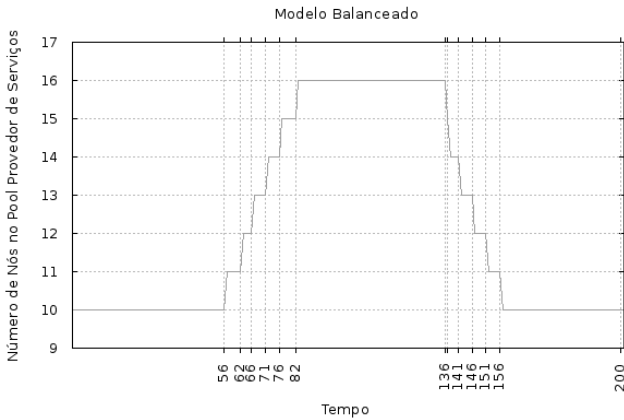


Figura 19: Número de nós no pool provedor de serviços no modelo balanceado em resposta aos recursos requisitados

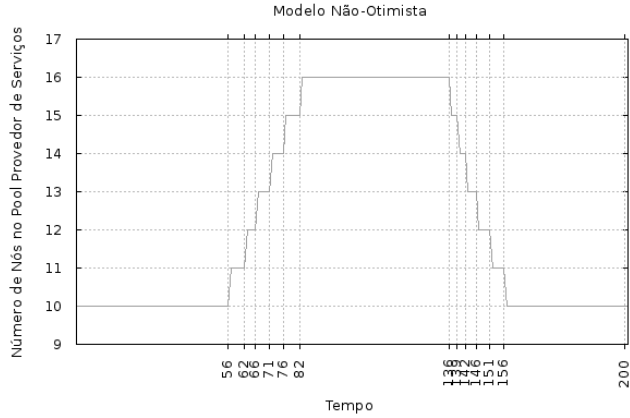


Figura 20: Número de nós no pool provedor de serviços no modelo não-otimista em resposta aos recursos requisitados

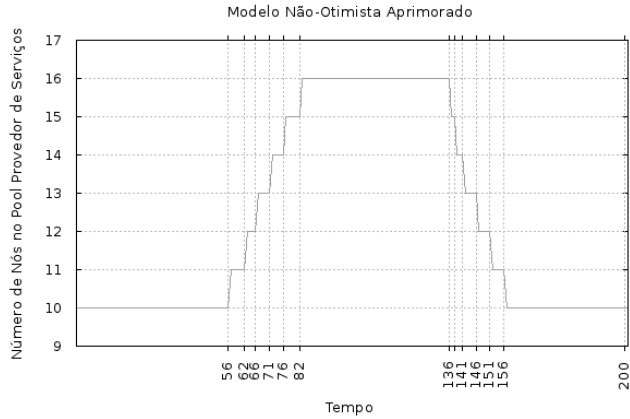


Figura 21: Número de nós no pool provedor de serviços no modelo não-otimista aprimorado em resposta aos recursos requisitados

É digno de nota que, ao se analisar os modelos *otimista* e *não-otimista* através de seus gráficos de variação de recursos individuais, nas figuras 14 e 16, pode-se perceber que os mesmos tem o comportamento especial de definir a capacidade de recursos individuais ao máximo no decremento do modelo *não-otimista*, e ao mínimo no incremento do

modelo *otimista*. Comportamento que aqui não chama tanto a atenção por estarmos lidando com nós com capacidade variando apenas de 1000 a 1200. E, apesar de sua utilidade, é um comportamento que, quando não considerado corretamente, pode ter sério impacto sobre serviços.

De maneira a termos o modelo balanceado sempre indicado como padrão.

5.2 LIMITAÇÕES E CASOS DE FALHA

Mesmo o modelo sendo dito “robusto o suficiente para se evitar situações de *acordar-adormecer* contínuos de nós”, é possível se simular casos extremos forçando tal comportamento.

A má definição de desvio padrão pode também ser a causa de tal comportamento, de maneira que é recomendável toda uma bateria de testes antes da tomada de decisão sobre como se definir tais valores. E é possível que, mesmo com a realização de testes para esta tomada de decisão, a existência de estados possíveis, não imaginados, capazes de gerar tal comportamento, seja alcançada.

É claro que estes seriam erros de *design* comportamental, e não erros do modelo em si, o qual estaria se comportando como o esperado, independente de se desejar ou não tal variação de número de nós.

Como este é um modelo desenvolvido para lidar explicitamente com erros relativos a nós, e em especial as suas falhas, a identificação das mesmas deve ser sanada com a substituição dos nós faltosos por outras réplicas do *pool de réplicas dormentes*.

O modelo se limita as avaliações como definidas em sua implementação, porém é genérico o suficiente para lidar com avaliações em várias dimensões, tentando sanar as mesmas.

Porém lidar com avaliações em várias dimensões, e tentar saná-las de maneira independente, pode levar a casos de *looping*, seja por *acréscimo-decrécimo* de recursos, ou mesmo casos mais complexos relativos a situações de *acordar-adormecer* de nós.

Este é um modelo que quando estendido a várias dimensões tem sua complexidade aumentada, e exige cuidado na formulação de sistemas lidando com tais restrições.

6 CONCLUSÕES E TRABALHOS FUTUROS

Esta tese iniciou introduzido o principal questionamento de pesquisa da mesma: a possibilidade de se avaliar *QoE* de maneira sistemática quando se lida com *BDDs*. Ela apresenta também a contextualização do maior problema a se definir, os conceitos abstratos de *qualidade* como percebidos do ponto de vista do *cliente* consumindo o serviço.

Sendo apontado que, quando lida-se com *BDDs* o *cliente* é, em geral, um *sistema* lidando com *ANSs*, e não o usual *usuário* consumindo serviços entregues na abordagem de *melhor esforço*, como embasado nos trabalhos originais abordando *QoS* com seu foco original em *serviços de rede*.

Em uma revisão conceitual são embasadas as concepções da referida hipótese, com foco na possibilidade de mapeamento sistemático de *QoE* através de *QoS* segundo *ANSs*. Os conceitos de *qualidade* são apresentados em suas origens, bem como as definições de *QoE* e *QoS*.

Através de revisão bibliométrica, é providenciada uma visão sistemática sobre o tópico de *QoE* em uma consideração mais genérica, e demonstrada a falta de pesquisa acadêmica sobre *QoE* focada em *arquiteturas de BDDs* através de indicadores estatísticos. Mesmo com o tópico de *QoE* estando em crescimento desde 2001, o mesmo parece não ter causado repercussão sobre a *área de pesquisa de BDDs*, demonstrando, assim, a inovação da pesquisa apresentada.

Os tópicos apresentados possuem muitas definições interessantes, além de várias publicações sendo apresentadas com o conceito já bem definido de *QoS* na área de pesquisa de redes, apesar das definições difusas de *QoE*. Entretanto, relativo à definição de *modelos QoE-QoS*, modelos palpáveis não foram encontrados, mesmo em publicações que mencionam a importância de explicitação de tal mapeamento.

Esta tese demonstra que o tópico de *QoE* atingiu seu pico de interesse de pesquisa em 2012, e que ainda é um tópico com muita pesquisa sendo realizada. Espera-se que, com base nestes dois trabalhos correlacionados, (COSTA; FURTADO, 2011) e (COSTA; FURTADO, 2013), e também na conceitualização aqui apresentada, novas pesquisas aplicando *QoE* a *arquiteturas de BDDs* sejam levadas adiante, como o modelo arquitetural apresentado, e mesmo a *arquiteturas distribuídas* em geral, como o modelo expandido apresentado em sequência.

Definições teóricas de um *modelo QoE-BDD* baseado em *requisitos de acesso a dados (RADs)* para definir *objetivos de nível de serviço (ONS)* foram brevemente apresentadas na revisão dos resultados bibli-

ométricos, além de uma maneira de se medir *QoS* com *indicadores de performance chave* (*IPCs*) encontrados em (COSTA; FURTADO, 2011), e citado comparações em alguns cenários apresentados em (COSTA; FURTADO, 2011) e (COSTA; FURTADO, 2013) dando suporte não apenas ao modelo apresentado, mas também a ideia de se atingir um melhor nível de *QoE* em *BDDs*.

O modelo definido por estes autores parece ser, além do modelo apresentado nesta tese, o único modelo proposto na literatura sobre *arquiteturas de BDDs* que enfatiza a implementação de *sistemas de QoE*, como demonstrado nos resultados bibliométricos estatísticos. E, mesmo com a demonstração de que *RADs* definindo *ONSs* medidos por *IPCs* parecem ter bons resultados nos trabalhos citados, os resultados são de difícil reprodução devido à falta de definição matemática explícita.

Apesar deste problema de reprodutibilidade, este modelo abriu novas possibilidades de consideração para *avaliação de QoE*. E mesmo considerando que o usuário pode explicitamente definir o comportamento esperado - um ponto que foi desconsiderado em relação a estes dois trabalhos mencionados -, os autores em questão demonstraram uma maneira de se iniciar a avaliação de serviços sobre *BDDs*.

Com fundamento nestas considerações de avaliação, apresenta-se, como resposta ao questionamento inicial sobre possibilidade de se avaliar *qualidade de experiência* em *bases de dados distribuídas*, não só definições de *QoS* e *QoE* e seu mapeamento, mas também arquiteturas de recuperação baseadas em completude de tempo e de recursos.

Com uma primeira base para o uso de *QoS* em *sistemas de BDDs* sendo definida para posterior apresentação de arquiteturas que permitem a criação de *sistemas autorrecuperáveis de QoE-QoS*. Demonstrando então a aplicação destas avaliações de *qualidade*, de maneira a responder o questionamento posterior de possibilidade de se prever e então evitar que estados inesperados sejam alcançados.

E, como uma implicação evidente deste primeiro modelo matemático bem definido, o caminho para *sistemas autorrecuperáveis de QoE-QoS* pode agora ser coberto, tendo um possível impacto futuro interessante sobre usuários e consumidores de serviços de *BDDs*, assim como sobre os provedores de serviços e a indústria de *BDs* em geral.

Os *parâmetros de QoS* analisados para concluir as *avaliações de QoE* foram apresentados de maneira a tentar demonstrar um modo de se correlacionar os conceitos de *QoE* com *QoS* lidando com *BDDs*.

Definições foram apresentadas para fundamentar um primeiro modelo de recuperação, dando garantias de *QoS* definidas em um *ANS*, com os intervalos de confiança de *QoE*, com posterior evolução deste

modelo para uma arquitetura de recuperação generalizada focada em recursos de sistema.

O modelo arquitetural inicial, lidando com completude de tempo, indicou a possibilidade de utilização de formulações complexas, de maneira a se lidar com avaliações como apresentadas nas definições de *QoE* e *QoS* da modelagem arquitetural e, desta maneira, possibilitando a capacidade de se predizer, com o intuito de evitar, comportamento inesperado de um sistema antes que o mesmo atinja estados inesperados com um modelo arquitetural de confiabilidade, disponibilidade e manutenibilidade.

As soluções propostas foram baseadas em escalabilidade horizontal e vertical aplicadas a *sistemas de BDDs*, podendo ser diretamente implementadas sobre um *SGBD* para evitar que tais estados sejam alcançados. Não sendo estes conceitos novidade *per se*, mas bastando para preencher este vazio conceitual sobre arquiteturas na área de bases de dados.

Um algoritmo simples, com alguns procedimentos considerando o comportamento temporal esperado, foi apresentado de maneira a ilustrar a sequência deste estudo, caminhando para a primeira implementação, então focada em completude de recursos.

Após a definição de modelagem arquitetural, finaliza-se com a apresentação de uma extensão mais generalizada, com a possibilidade de aplicação à *sistemas distribuídos*, agora com base em completude de recursos.

Nesta generalização, quatro modos de operação são apresentados para se lidar com um modelo de recuperação arquitetural focado em *sistemas distribuídos*, cada um dos mesmos com suas especificações e usos particulares. A decisão sobre qual modo se utilizar devendo envolver conhecimento do sistema sobre o qual se deseja aplicar tal modelo, sendo recomendado testes de performance sobre o sistema para verificar situações específicas e como cada um dos modelos irá se comportar.

E com o modelo e arquiteturas apresentadas, partimos para definição de um simulador que permitiu considerações de sistemas distribuídos sem a necessidade de construção de um ambiente de tal porte, de maneira a possibilitar a demonstração de ambientes utilizando as avaliações apresentadas para prever e evitar que estados inesperados fossem alcançados.

Durante a análise experimental, dados para uma necessidade específica de incremento-decremento foram apresentados, sendo também apontado que este é um comportamento não usual para um *sistema distribuído*, de maneira a explicar particularidades de cada um dos modos

apresentados.

Desta maneira partimos da hipótese de que poderíamos lidar com o mapeamento de *QoE* através da definição de *QoS*, com um primeiro modelo apresentado, e aplicamos sobre um modelo arquitetural. Este modelo foi então estendido para uma abordagem mais generalizada, com possibilidades comportamentais apresentadas em quatro modos de operação. Apresenta-se a análise experimental mostrando a possibilidade de predição comportamental para se evitar estados inesperados utilizando o ambiente de experimentação, comprovando não apenas a hipótese inicial, mas também demonstrando sua utilização.

O passo futuro natural é a construção de sistemas lidando com as três dimensões: capacidade de processamento, memória volátil e capacidade de armazenamento, se não as três combinadas, sobre sistemas distribuídos, com foco especial sobre *bases de dados distribuídas*.

Os resultados das definições estruturais, bem como dos dois modelos de recuperação propostos, foram publicados e tem seus detalhes contidos no anexo A.

REFERÊNCIAS

- ABADI, D. J. Consistency tradeoffs in modern distributed database system design. In: **Computer**. [S.l.]: IEEE, 2012. v. 45, p. 37–42. ISSN 0018-9162.
- ABRAMOWICZ, W. et al. Architecture for service profiling. In: **Services Computing Workshops**. Chicago, USA: [s.n.], 2006.
- ALVAREZ, A. et al. A flexible qoe framework for video streaming services. In: **IEEE GLOBECOM Workshops (GC Wkshps)**. Houston, TX: IEEE, 2011. p. 1226–1230. ISBN 978-1-4673-0040-7.
- ANGELOPOULOS, J. et al. Dynamic qos provisioning for ethernet-based networks. In: **International Electronic Conference on Computer Science**. Electr Network: Amer Inst Physics, 2008. (AIP Conference Proceedings, v. 1060), p. 151–154. ISBN 978-0-7354-0590-5. ISSN 0094-243X.
- BALDUZZI, D.; TONONI, G. Qualia: The geometry of integrated information. In: **PLoS Computational Biology**. [S.l.: s.n.], 2009. v. 5, n. 8. ISSN 1553734X.
- BARKOWSKY, M. et al. Towards standardized 3dtv qoe assessment: Cross-lab study on display technology and viewing environment parameters. In: **Proceedings of SPIE - The International Society for Optical Engineering**. Burlingame, CA; United States: [s.n.], 2013. v. 8648. ISBN 978-081949421-4. ISSN 0277786X.
- BARLA, I. B.; SCHUPKE, D. A.; CARLE, G. Resilient virtual network design for end-to-end cloud services. In: **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**. Prague; Czech Republic: [s.n.], 2012. v. 7289, n. Part I, p. 161–174. ISBN 978-364230044-8. ISSN 03029743.
- BARLA, I. B. et al. Optimal design of virtual networks for resilient cloud services. In: **2013 9th International Conference on the Design of Reliable Communication Networks, DRCN 2013**. [S.l.: s.n.], 2013. p. 218–225. ISBN 978-147990049-7.
- BHATTI, N.; BOUCH, A.; KUCHINSKY, A. Integrating user-perceived quality into web server design. [S.l.], 2000.

BOUCH, A.; BHATTI, N.; KUCHINSKY, A. **Quality is the eye of the beholder: Meeting user's requirements for internet quality of service.** [S.l.], 2000.

BROOKS, P.; HESTNES, B. User measures of quality of experience: why being objective and quantitative is important. In: **IEEE Network.** [S.l.]: IEEE, 2010. v. 24, n. 2, p. 8–13. ISSN 0890-8044.

CALLET, P. L.; MÖLLER, S.; PERKIS, A. Qualinet white paper on definitions of quality of experience. In: LAUSANNE (Ed.). **European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003).** Switzerland: [s.n.], 2013. Version 1.2.

CAO, Y.; RITZ, C.; RAAD, R. How much longer to go? the influence of waiting time and progress indicators on quality of experience for mobile visual search applied to print media. In: **2013 5th International Workshop on Quality of Multimedia Experience, QoMEX 2013 - Proceedings.** [S.l.: s.n.], 2013. p. 112–117.

CARUANA, A. Service loyalty: The effects of service quality and the mediating role of customer satisfaction. **European Journal of Marketing**, v. 36, n. 7/8, p. 811–828, 2002. ISSN 0309-0566.

CHANG, D. et al. A network-aware virtual machine placement algorithm in mobile cloud computing environment. In: **2013 IEEE Wireless Communications and Networking Conference Workshops, WCNCW 2013.** Shanghai; China: [s.n.], 2013. p. 117–121. ISSN 978-147990110-4.

CHEN, C. et al. A dynamic system model of time-varying subjective quality of video streams over http. In: **ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings.** Vancouver, BC; Canada: [s.n.], 2013. p. 3602–3606. ISBN 978-147990356-6. ISSN 15206149.

CLOUD, F. Y. **The definition of customer quality of experience.** August 2013. Online website. Available at <https://www.findyourcloud.com/Articles/Quality-of-Experience/The-Definition-of-Customer-Quality-of-Experience/>.

CLOUD, F. Y. **Cloud computing & service quality articles from find your cloud.** February 2014. Online website. Available at <https://www.findyourcloud.com/Articles/Quality-of-Experience/>.

COSTA, R. L. de C.; FURTADO, P. Quality of experience in distributed databases. In: NETHERLANDS, S. (Ed.). **Distributed and Parallel Databases**. 5-6. ed. [S.l.: s.n.], 2011. v. 29, p. 361–396.

COSTA, R. L. de C.; FURTADO, P. Providing quality of experience for users: The next dbms challenge. In: SOC, I. C. (Ed.). **Computer**. 9. ed. [S.l.: s.n.], 2013. v. 46, p. 86–93.

CROTTI, M. et al. Pp2db: A privacy-preserving, p2p-based scalable storage system for mobile networks. In: **Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering**. London; United Kingdom: [s.n.], 2011. v. 96, p. 533–542. ISBN 978-364231908-2. ISSN 18678211.

D'AGOSTINO, R. B. Transformation to normality of the null distribution of g_1 . In: **Biometrika** **57** (3). [S.l.: s.n.], 1970. p. 679–681.

DATABASES, N. January 2014. Online website. Available at <http://nosql-database.org/>.

ELSEVIER, B. V. **Scopus**. November 2013. Online website. Available at <http://www.scopus.com>.

ERL, T.; PUTTINI, R.; MAHMOOD, Z. **Cloud Computing: Concepts, Technology & Architecture**. 1. ed. [S.l.]: Prentice Hall, 2013. (The Prentice Hall Service Technology Series from Thomas Erl).

FIGIORESE, A.; SIMES, P.; BOAVIDA, F. An approach to peer selection in service overlays. In: **7th International Conference on Network and Service Management, CNSM 2011**. Paris; France: [s.n.], 2011. ISBN 978-145771588-4.

FIGIORESE, A.; SIMOES, P.; BOAVIDA, F. Peer selection in p2p service overlays using geographical location criteria. In: **Computational Science and Its Applications - 12th International Conference on Computational Science and Its Applications (ICCSA)**. Salvador de Bahia, BRAZIL: SPRINGER-VERLAG BERLIN, 2012. (Lecture Notes in Computer Science, v. 7334), p. 234–248. ISBN 978-3-642-31074-4, 978-3-642-31075-1. ISSN 0302-9743.

FIGIORESE, A.; SIMOES, P.; BOAVIDA, F. Approach for service search and peer selection in p2p service overlays. In: **27th International Conference on Information Networking**

(**ICOIN**). Bangkok, THAILAND: IEEE, 2013. ISBN 978-1-4673-5742-5, 978-1-4673-5740-1.

FLIEGEL, K. et al. Comparison of stereoscopic technologies in various configurations. In: **Conference on Applications of Digital Image Processing XXXV**. San Diego, CA: Spie-Int Soc Optical Engineering, 2012. (Proceedings of SPIE, v. 8499). ISSN 0277-786X.

FLORIS, A. et al. Qoe assessment of multimedia video consumption on tablet devices. In: **IEEE Globecom Workshops (GC Wkshps)**. Anaheim, CA: IEEE, 2012. (IEEE Globecom Workshops). ISBN 978-1-4673-4941-3. ISSN 2166-0069.

FORLIZZI, J.; FORD, S. Building blocks of experience: An early framework for interaction designers. In: **Proceedings of the Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques, DIS**. [S.l.]: ACM, New York, NY, United States, 2000. p. 419–423.

FORTETSANAKIS, G. et al. Supporting wireless access markets with a user-centric qoe-based geo-database. In: . Istanbul; Turkey: [s.n.], 2012. p. 29–36. ISBN 978-145031526-5.

GILBERT, S.; LYNCH, N. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. In: **ACM SIGACT**. [S.l.]: ACM New York, NY, USA, 2002. v. 33, p. 51–59.

GOLDMANN, L.; LEE, J.-S.; EBRAHIMI, T. Temporal synchronization in stereoscopic video: Influence on quality of experience and automatic asynchrony detection. In: **IEEE International Conference on Image Processing**. Hong Kong, PEOPLES R CHINA: IEEE, 2010. p. 3241–3244. ISSN 1522-4880.

GOLDMANN, L.; SIMONE, F. D.; EBRAHIMI, T. A comprehensive database and subjective evaluation methodology for quality of experience in stereoscopic video. In: **Proceedings of SPIE - The International Society for Optical Engineering**. San Jose, CA; United States: [s.n.], 2010. v. 7526. ISBN 978-081947919-8. ISSN 0277786X.

GONG, Y. et al. Model-based approach to measuring quality of experience. In: **First International Conference on Emerging Network Intelligence**. Sliema: IEEE, 2009. p. 29 – 32. ISBN 978-0-7695-3835-8.

GRONROOS, C. A service quality model and its market implications. **European Journal of Marketing**, v. 18, n. 4, p. 36–44, 1984.

HE, Y. et al. Pnn-based qoe measuring model for video applications over lte system. In: **7th International ICST Conference on Communications and Networking in China (CHINACOM)**. Kunming, Peoples R China: IEEE, 2012. ISBN 978-1-4673-2698-8.

HEIMBIGNER, D.; MCLEOD, D. A federated architecture for information management. In: **ACM. ACM Transactions on Information Systems (TOIS)**. New York, NY, USA, 1985. v. 3, n. 3, p. 253–278.

HINES, A. et al. Robustness of speech quality metrics to background noise and network degradations: Comparing visqol, pesq and polqa. In: **ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings**. Vancouver, BC; Canada: [s.n.], 2013. p. 3697–3701. ISBN 978-147990356-6. ISSN 15206149.

HONDA, Y.; NINOMIYA, H.; SASSA, Y. Development of cabin air quality system. In: **SAE 2001 World Congress**. Detroit, MI; United States: [s.n.], 2001.

HOOGEBOOM, K. R. Restoration and development guidelines for ocean beach recreation areas. In: **Coastal Zone '89: Proceedings of the Sixth Symposium on Coastal and Ocean Management**. Charleston SC USA: Publ by ASCE, New York, NY, United States, 1989. v. 4, n. pt4, p. 3120–3134.

IC-NIDC. **Proceedings - 2012 3rd IEEE International Conference on Network Infrastructure and Digital Content, IC-NIDC 2012**. September 2012. Beijing; China.

ICETE. **10th Int. Joint Conf. on ICETE 2013; 4th Int. Conf. on DCNET 2013, 10th Int. Conf. on ICE-B 2013 and OPTICS 2013 - 4th Int. Conf. on Optical Communication Systems**. July 2013. Reykjavik; Iceland.

II, A. T.; NOWELL, L. T. Beyond the desktop: Diversity and artistry. In: **Conference on Human Factors in Computing Systems, CHI EA 2000**. [S.l.: s.n.], 2000. p. 35–36. ISBN 1581132484;978-158113248-9.

ISO9241-210. **ISO 9241-210:2010 - Human-centred design processes for interactive systems**. 2010. Available at http://www.iso.org/iso/catalogue_detail.htm?csnumber=21197.

ITU-T.P.862. **Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs**. Geneva, Switzerland, 2001.

JARQUE, C. M.; BERA, A. K. Efficient tests for normality, homoscedasticity and serial independence of regression residuals. In: **Economics Letters 6 (3)**. [S.l.: s.n.], 1980. p. 255–259.

JELASSI, S.; RUBINO, G. A comparison study of automatic speech quality assessors sensitive to packet loss burstiness. In: **IEEE Consumer Communications and Networking Conference, CCNC'2011**. Las Vegas, NV; United State: [s.n.], 2011. p. 415–420. ISBN 978-142448790-5.

JELASSI, S.; RUBINO, G. A study of artificial speech quality assessors of voip calls subject to limited bursty packet losses. In: **Eurasip Journal on Image and Video Processing**. [S.l.]: Springer International Publishing AG, 2011. ISSN 1687-5281.

KHAN, A.; SUN, L.; IFEACHOR, E. Qoe prediction model and its application in video quality adaptation over umts networks. In: **IEEE TRANSACTIONS ON MULTIMEDIA**. [S.l.]: IEEE-Inst Electrical Electronics Engineers Inc, 2012. v. 14, n. 2, p. 431–442. ISSN 1520-9210.

KIM, H.-J. et al. The qoe evaluation method through the qos-qoe correlation model. In: **Fourth International Conference on Networked Computing and Advanced Information Management. NCM '08**. Gyeongju: IEEE, 2008. v. 2, p. 719 – 725. ISBN 978-0-7695-3322-3. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4624234>>.

LAGHARI, A. A. et al. Qon: Quality of experience (qoe) framework for network services. In: **4th International Conference on Software Technology and Engineering (ICSTE 2012)**. Univ Madras, Phuket, THAILAND: Amer Soc Mechanical Engineers, 2012. ISBN 978-0-7918-6015-1.

LAGHARI, K. ur R.; CRESPI, N.; CONNELLY, K. Towards total quality of experience: A qoe for multimedia services in communication ecosystem. **IEEE Communication Magazine**, April 2012.

LAGHARI, K. ur R. et al. Qoe aware service delivery in distributed environment. In: **Advanced Information Networking and Applications (AINA), 2011 IEEE Workshops of International Conference on**. [S.l.: s.n.], 2011. p. 837 – 842. ISBN 978-1-61284-829-7. Available at <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5763609>.

LEE, J.-S. Comparison of objective quality metrics on the scalable extension of h.264/avc. In: **Proceedings - International Conference on Image Processing, ICIP**. Lake Buena Vista, FL; United States: [s.n.], 2012. p. 693–696. ISBN 978-146732533-2. ISSN 15224880.

LEE, J.-S.; GOLDMANN, L.; EBRAHIMI, T. Paired comparison-based subjective quality assessment of stereoscopic images. In: **Multimedia Tools and Applications**. [S.l.]: SPRINGER, 2013. v. 67, n. 1, p. 31–48. ISSN 1380-7501.

LEWIS, R. C.; BOOMS, B. H. The marketing aspects of service quality. In: BERRY, L. L.; SHOSTACK, G.; UPAH, G. (Ed.). **Emerging Perspectives in Service Marketing**. Chicago, IL. USA., 1983. p. 99–107.

LI, M.; CHEN, Z.; TAN, Y.-P. Qoe analysis for scalable video adaptation. In: **IEEE Visual Communications and Image Processing (VCIP)**. San Diego, CA: IEEE, 2012. ISBN 978-1-4673-4405-0.

LI, M.; CHEN, Z.; TAN, Y.-P. Qoe-aware resource allocation for scalable video transmission over multiuser mimo-ofdm systems. In: **IEEE Visual Communications and Image Processing (VCIP)**. San Diego, CA: IEEE, 2012. ISBN 978-1-4673-4405-0.

LI, M.; CHEN, Z.; TAN, Y.-P. On quality of experience of scalable video adaptation. In: . [S.l.]: ACADEMIC PRESS INC, 2013. v. 24, n. 5, p. 509–521. ISSN 1047-3203.

LIU, X.; SUN, C.; YANG, L. T. Dct-based objective quality assessment metric of 2d/3d image. In: **Multimedia Tools and Applications**. [S.l.: s.n.], 2013. p. 1–18. ISSN 13807501.

LOUKIL, M. et al. A reputation based vertical handover decision making framework (r-vhdf). In: **IEEE Globecom Workshops (GC Wkshps)**. [S.l.]: IEEE, 2012. (IEEE Globecom Workshops). ISBN 978-1-4673-4941-3. ISSN 2166-0069.

MAHDI, A. E.; PICOVICI, D. New single-ended objective measure for non-intrusive speech quality evaluation. In: **Signal Image and Video Processing**. [S.l.]: Springer London Ltd, 2010. v. 4, n. 1, p. 23–38. ISSN 1863-1703.

MAI, Z. et al. Rendering 3-d high dynamic range images: Subjective evaluation of tone-mapping methods and preferred 3-d image attributes. In: **IEEE Journal of Selected Topics in Signal Processing**. [S.l.]: IEEE-Inst Electrical Electronics Engineers Inc, 2012. v. 6, n. 5, p. 597–610. ISSN 1932-4553.

MARANDI, P. J.; PRIMI, M.; PEDONE, F. Multi-ring paxos. In: **Conference on Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International**. Boston, MA: IEEE, 2012. ISBN 978-1-4673-1624-8. ISSN 1530-0889.

MIELKE, R. et al. Simulation applied to theme park management. In: IEEE. **Proceedings of the 1998 Winter Simulation Conference, WSC. Part 1 (of 2)**. Washington DC USA: IEEE, Piscataway, NJ, United States, 1998. v. 2, p. 1199–1203. ISSN 02750708.

MOBIARCH'12. Mobiarch'12 - 7th acm international workshop on mobility in the evolving internet architecture. In: **Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM**. Istanbul; Turkey: [s.n.], 2012. ISBN 978-145031526-5.

MÖLLER, S. et al. Comparison of approaches for instrumentally predicting the quality of text-to-speech systems. In: **11th Annual Conference of the International-Speech-Communication-Association 2010**. Makuhari, JAPAN: Isca-Inst Speech Communication Assoc, 2010. v. 1-2, p. 1325–1328. ISBN 978-1-61782-123-3.

MÖLLER, S.; RAAKE, A. (Ed.). **Quality of Experience: Advanced Concepts, Applications and Methods**. 1. ed. [S.l.]: Springer International Publishing, 2014. ISSN 2192-2810. ISBN 978-3-319-02681-7.

MOOR, K. D. et al. Proposed framework for evaluating quality of experience in a mobile, testbed-oriented living lab setting. In: **Mobile Networks & Applications**. [S.l.]: Springer, 2010. v. 15, n. 3, p. 378–391. ISSN 1383-469X.

MOORTHY, A. K.; BOVIK, A. C. Visual quality assessment algorithms: What does the future hold? In: **Multimedia Tools and Applications**. [S.l.: s.n.], 2011. v. 51, n. 2, p. 675–696. ISSN 13807501.

MOORTHY, A. K.; BOVIK, A. C. A survey on 3d quality of experience and 3d quality assessment. In: **Proceedings of SPIE - The International Society for Optical Engineering**. Burlingame, CA; United States: [s.n.], 2013. v. 8651. ISBN 978-081949424-5. ISSN 0277786X.

MOORTHY, A. K. et al. Subjective evaluation of stereoscopic image quality. In: **SIGNAL PROCESSING-IMAGE COMMUNICATION**. [S.l.]: ELSEVIER SCIENCE BV, 2013. v. 28, n. 8, p. 870–883. ISSN 0923-5965.

MORRIS, M. G.; TURNER, J. M. Assessing users' subjective quality of experience with the world wide web: an exploratory examination of temporal changes in technology acceptance. **International Journal of Human-Computer Studies**, v. 54, n. 6, p. 877–901, June 2001. ISSN 1071-5819.

MOY, C.; DOYLE, L.; SANADA, Y. Foreword-cognitive radio: From equipment to networks. In: **Annales des Telecommunications/Annals of Telecommunications**. [S.l.: s.n.], 2009. v. 64, n. 7-8, p. 415–417. ISSN 00034347.

MUSHTAQ, M. S.; AUGUSTIN, B.; MELLOUK, A. Empirical study based on machine learning approach to assess the qos/qoe correlation. In: **17th European Conference on Network and Optical Communications, NOC 2012, 7th Conference on Optical Cabling and Infrastructure, OC and I 2012**. Vilanova i la Geltru; Spain: [s.n.], 2012. ISBN 978-146730951-6.

NORSIG. **Proceedings of the 7th Nordic Signal Processing Symposium, NORSIG 2006**. June 2006. Reykjavik; Iceland.

PAPAFILI, I. et al. Smartenit cloud traffic management approach and architectural considerations. In: **Future Network and Mobile**

Summit, FutureNetworkSummit 2013. Lisbon, Portugal: [s.n.], 2013. ISBN 978-190582437-.

PAPAZOGLU, M. P.; BUBENKO, J.; NORRIE, M. Intent: An integrated environment for distributed heterogeneous databases. In: **ACM. OOPSLA/ECOOP '88 Proceedings of the 1988 ACM SIGPLAN - workshop on Object-based concurrent programming.** New York, NY, USA, 1988. p. 126–128. ISBN 0-89791-304-3.

PARASURAMAN, A.; ZEITHAML, V. A.; BERY, L. L. A conceptual model of service quality and its implication for future research. **Journal of Marketing**, v. 49, p. 41–50, April 1985.

PARASURAMAN, A.; ZEITHAML, V. A.; BERY, L. L. Servqual: a multiple-item scale for measuring consumer perceptions of service quality. **Journal of Retailing**, v. 64, n. 1, p. 12–40, 1988.

PARASURAMAN, A.; ZEITHAML, V. A.; BERY, L. L. Alternative scales for measuring service quality: a comparative assessment based on psychometric and diagnostic criteria. **Journal of Retailing**, v. 70, n. 3, p. 201–30, 1994.

QI, F. et al. Quality of experience assessment for stereoscopic images. In: **IEEE International Symposium on Circuits and Systems.** Seoul, SOUTH KOREA: IEEE, 2012. p. 1712–1715. ISSN 0271-4302.

QOMEX. **4th International Workshop on Quality of Multimedia Experience, QoMEX 2012.** July 2012. Melbourne, VIC; Australia.

REDD, E. et al. Measurements of helium excitation in be+–he and mg+–he collisions. In: **Nuclear Inst. and Methods in Physics Research, B.** [S.l.: s.n.], 1987. v. 24-25, n. PART 1, p. 305–308. ISSN 0168583X.

REHMAN, A.; WANG, Z. Perceptual experience of time-varying video quality. In: **2013 5th International Workshop on Quality of Multimedia Experience, QoMEX 2013 - Proceedings.** Klagenfurt am Worthersee; Austria: [s.n.], 2013. p. 218–223.

REUTERS, T. **Web of Knowledge at Web of Science v.5.1.3.1.** November 2013. Online website. Available at <http://apps.webofknowledge.com>.

RICHTER, S.; BOHM, A. A location and privacy service enabler for context-aware and location-based services in ngn. In: **Telecommunications Network Strategy and Planning Symposium**. New Delhi, India: [s.n.], 2006.

SATO, K.; VERPLANK, W. Panel: Teaching tangible interaction design. In: **Proceedings of the Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques, DIS**. [S.l.]: ACM, New York, NY, United States, 2000. p. 444–445.

SEDANO, I. et al. Evaluation of video quality metrics on transmission distortions in h.264 coded video. In: **IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, BMSB 2011 - Conference Programme**. Nuremberg, Germany: [s.n.], 2011. ISBN 978-161284122-9.

SHAPIRO, S. S.; WILK, M. B. An analysis of variance test for normality (complete samples). In: **Biometrika** **52 (3-4)**. [S.l.: s.n.], 1965. p. 591–611.

SHENKER, S.; PARTRIDGE, C.; GUERIN, R. **RFC 2212: Specification of Guaranteed Quality of Service**. [S.l.], September 1997. Disponível em: <http://tools.ietf.org/html/rfc2212>.

SHENKER, S.; WROCLAWSKI, J. **RFC 2216: Network Element Service Specification Template**. [S.l.], September 1997. Disponível em: <http://tools.ietf.org/html/rfc2216>.

SHI, Y.; DING, Y.; LI, J. A user-perceived video quality assessment metric using inter-frame redundancy. In: **International Journal of Digital Content Technology and its Applications**. [S.l.: s.n.], 2010. v. 4, n. 4, p. 151–160. ISSN 19759339.

SIEWIOREK, D. P.; SWARZ, R. S. **Reliable computer systems: design and evaluation**. 3th. ed. Natick, Mass. : A K Peters, 1998. P. 508. ISBN 1-56881-092-X. Disponível em: <http://archive.org/details/reliablecomputer00siew>.

SLADOJEVIC, S. et al. Logging real packet reception patterns for end-to-end quality of experience assessment in wireless multimedia transmission. In: **Electronic Proceedings of the 2013 IEEE International Conference on Multimedia and Expo**

Workshops, ICMEW 2013. San Jose, CA; United States: [s.n.], 2013. ISBN 978-147991604-7.

SMITH, J. M. et al. Multibase: Integrating heterogeneous distributed database systems. In: ACM. **AFIPS National Computer Conference.** New York, NY, USA, 1981.

SOLDANI, D.; LI, M.; CUNY, R. **QoS and QoE management in UMTS cellular systems.** [S.l.]: Wiley, 2006.

SOUZA, R. H. de; DANTAS, M. A. R. Mapping QoE through QoS in an approach to DDB architectures: Research analysis and conceptualization. **ACM Computing Surveys (CSUR)**, v. 48, n. 2, November 2015.

SOUZA, R. H. de et al. Architectural recovering model for distributed databases: A reliability, availability and serviceability approach. In: **2016 IEEE Symposium on Computers and Communication (ISCC).** Messina, Italy: [s.n.], 2016. p. 575–580. ISBN 978-1-5090-0680-9.

SPIE-IS. Proceedings of spie-is and t electronic imaging - multimedia on mobile devices 2011; and multimedia content access: Algorithms and systems v. In: **Proceedings of SPIE - The International Society for Optical Engineering.** San Francisco, CA; United States: [s.n.], 2011. v. 7881. ISBN 978-081948418-5. ISSN 0277786X.

STAELENS, N. et al. High definition h.264/avc subjective video database for evaluating the influence of slice losses on quality perception. In: . Klagenfurt am Worthersee; Austria: [s.n.], 2013. p. 130–135.

TAHERI, M.; ANSARI, N. A feasible solution to provide cloud computing over optical networks. In: **IEEE Network.** [S.l.: s.n.], 2013. v. 27, n. 6, p. 31–35. ISSN 08908044.

TRAN, H. A.; MELLOUK, A.; HOCEINI, S. Qoe content distribution network for cloud architecture. In: **Proceedings - 2011 1st International Symposium on Network Cloud Computing and Applications, NCCA 2011.** Toulouse; France: [s.n.], 2011. p. 14–19. ISBN 978-076954550-9.

VANDENBROUCKE, K.; MOOR, K. D.; MAREZ, L. D. Use- and qoe-related aspects of personal cloud applications: An exploratory survey. In: **Quality of Multimedia Experience (QoMEX), 2013**

Fifth International Workshop on. Klagenfurt am WÃ¶rthersee: [s.n.], 2013. p. 36–37. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6603202>.

VERA, D. D.; RODRIGUEZ-BOCCA, P.; RUBINO, G. Qoe monitoring platform for video delivery networks. In: **7th IEEE International Workshop on IP Operations and Management**. San Jose, CA: Springer-Verlag Berlin, 2007. (Lecture Notes in Computer Science, v. 4786), p. 131–142. ISBN 978-3-540-75852-5. ISSN 0302-9743.

WEI, Y.; SON, S. H.; STANKOVIC, J. A. Maintaining data freshness in distributed real-time databases. In: **16th Euromicro Conference on Real-Time Systems. ECRTS 2004**. [S.l.]: IEEE, 2004. p. 251 – 260. ISBN 0-7695-2176-2. ISSN 1068-3070.

WIJNANTS, M. et al. Optimizing user qoe through overlay routing, bandwidth management and dynamic transcoding. In: **IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks**. Newport Beach, CA: IEEE, 2008. v. 1-2, p. 557–563. ISBN 978-1-4244-2099-5.

WOLTER, K.; MOORSEL, A. van. **The relationship between quality of service and business metrics: Monitoring, notification and optimization**. [S.l.], April 2001.

XING, L. et al. Estimating quality of experience on stereoscopic images. In: **ISPACS 2010 - 2010 International Symposium on Intelligent Signal Processing and Communication Systems, Proceedings**. Chengdu; China: [s.n.], 2010. ISBN 978-142447371-7.

XING, L. et al. An objective metric for assessing quality of experience on stereoscopic images. In: **IEEE International Workshop on Multimedia Signal Processing, MMSP2010**. Saint Malo; France: [s.n.], 2010. p. 373–378. ISBN 978-142448112-5.

XING, L. et al. Objective metrics for quality of experience in stereoscopic images. In: **18th IEEE International Conference on Image Processing (ICIP)**. Brussels, BELGIUM: IEEE, 2011. ISBN 978-1-4577-1303-3. ISSN 1522-4880.

XU, D.; CORIA, L.; NASIOPOULOS, P. Guidelines for capturing high quality stereoscopic content based on a systematic subjective evaluation. In: **IEEE International Conference on Electronics,**

Circuits, and Systems, ICECS 2010 - Proceedings. Athens; Greece: [s.n.], 2010. p. 162–165. ISBN 978-142448157-6.

XU, D.; CORIA, L. E.; NASIOPOULOS, P. Guidelines for an improved quality of experience in 3-d tv and 3-d mobile displays. In: **JOURNAL OF THE SOCIETY FOR INFORMATION DISPLAY**. [S.l.]: SOC INFORMATION DISPLAY, 2012. v. 20, n. 7, p. 397–407. ISSN 1071-0922.

YU, K. et al. Analysis of overlay topology of peer-to-peer applications. In: **Journal of China Universities of Posts and Telecommunications**. [S.l.: s.n.], 2010. v. 17, n. SUPPL. 2, p. 60–66. ISSN 10058885.

ZERMAN, E. et al. Spatiotemporal no-reference video quality assessment model on distortions based on encoding. In: **21st Signal Processing and Communications Applications Conference (SIU)**. [S.l.]: IEEE, 2013. (Signal Processing and Communications Applications Conference). ISBN 978-1-4673-5563-6, 978-1-4673-5562-9. ISSN 2165-0608.

ZERMAN, E. et al. Spatiotemporal no-reference video quality assessment model on distortions based on encoding. In: **21st Signal Processing and Communications Applications Conference, SIU 2013**. Haspolat; Turkey: [s.n.], 2013. ISBN 978-146735562-9.

ANEXO A – Trabalhos Publicados

Os trabalhos publicados durante a realização desta tese de doutorado são listados na tabela 19 em ordem cronológica, consistindo de 2 artigos: O primeiro, *Mapping QoE through QoS in an approach to DDB architectures: Research analysis and conceptualization*, publicado junto ao periódico internacional *ACM Computing Surveys (CSUR)*, em 2015; e o subsequente, *Architectural Recovering Model for Distributed Databases: A Reliability, Availability and Serviceability Approach*, publicado junto à conferência internacional *The Twenty-First IEEE Symposium on Computers and Communications (ISCC)*, em 2016, realizada em Messina, na Itália.

Artigos Publicados	
Título do Artigo:	Mapping QoE through QoS in an approach to DDB architectures: Research analysis and conceptualization
Veículo Publicado:	ACM Computing Surveys
Sigla Ano:	CSUR 2015
Tipo de Veículo:	Periódico Internacional
Autores:	SOUZA, R. H. e DANTAS, M. A. R.
Avaliação Capes:	A1
Status:	Publicado
Título do Artigo:	Architectural Recovering Model for Distributed Databases: A Reliability, Availability and Serviceability Approach
Veículo Publicado:	The Twenty-First IEEE Symposium on Computers and Communications
Sigla Ano:	ISCC 2016
Tipo de Veículo:	Congresso Internacional
Autores:	SOUZA, R. H., FLORES, P. A., DANTAS, M. A. R. e SIQUEIRA, F.
Avaliação Capes:	A2
Status:	Publicado

Tabela 19: Artigos publicados em ordem cronológica.

O primeiro dos artigos, *Mapping QoE through QoS in an approach to DDB architectures: Research analysis and conceptualization* (SOUZA; DANTAS, 2015), consiste da base essencial que culminou no desenvolvimento desta tese, com os primeiros conceitos de *QoS* para *BDDs* formalizados, permitindo então seu mapeamento direcionado à métricas de *QoE*.

Este trabalho inclui também estudo bibliométrico demonstrando, com a utilização de indicadores estatísticos, a inexistência de tais conceitos à área até o momento de início de formalização do mesmo, bem como sua relevância.

O segundo dos artigos, *Architectural Recovering Model for Distributed Databases: A Reliability, Availability and Serviceability Approach* (SOUZA et al., 2016), consiste de uma primeira implementação dos conceitos apresentados junto à publicação anterior, de maneira a demonstrar as primeiras possibilidades de se lidar com os conceitos como definidos.

Este modelo lida apenas com as definições explícitas de *RAS*, de maneira que as avaliações de *R&Q* ainda precisam ser considera-

das para avaliação sistemática de sistemas. Ele cobre mais a análise e avaliação comportamental a longo prazo, não se focando em falhas pontuais, que devem ser avaliadas por análise do modelo de equações REQ .

Um terceiro artigo, intitulado *An Architectural Recovering Model for Distributed Systems' Resources*, surgido como uma evolução natural do sistema de avaliação aplicável à sistemas distribuídos em geral, com foco em recursos de sistemas, será submetido após termino desta tese a congresso com avaliação Capes A. Esta abordagem surge da realização da análise experimental da tese, como uma abordagem a permitir uma avaliação genérica, independente de sistemas específicos, demonstrando o ganho que pode ser obtido com a aplicação desta abordagem híbrida de QoE como completude de QoS .